WL-TR-97-3078

INTERFACE-DRIVEN MULTIDISCIPLINARY DESIGN OF LARGE-SCALE AIRCRAFT STRUCTURES

DAVID R. OAKLEY
GRAHAM S. RHODES
LONNY B. KRUGER

APPLIED RESEARCH ASSOCIATES, INC.
811 SPRING FOREST ROAD, SUITE 100
RALEIGH NC 27609

FEBRUARY 1997

FINAL DRAFT REPORT FOR 1 DEC 95 - 1 DEC 96

Approved for public release; distribution unlimited

FLIGHT DYNAMICS DIRECTORATE
WRIGHT LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AFB OH 45433-7562

NOTICE

USING GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA INCLUDED IN THIS DOCUMENT FOR ANY PURPOSE OTHER THAN GOVERNMENT PROCUREMENT DOES NOT IN ANY WAY OBLIGATE THE US GOVERNMENT. THE FACT THAT THE GOVERNMENT FORMULATED OR SUPPLIED THE DRAWINGS, SPECIFICATIONS, OR OTHER DATA DOES NOT LICENSE THE HOLDER OR ANY OTHER PERSON OR CORPORATION; OR CONVEY ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY RELATE TO THEM.

THIS REPORT IS RELEASABLE TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS). AT NTIS, IT WILL BE AVAILABLE TO THE GENERAL PUBLIC, INCLUDING FOREIGN NATIONS.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

Max U Blair

Dr. Max Blair, Aerospace Engineer
WL/FIBD

Nelson D. Wolf

Nelson D. Wolf, Branch Chief
WL/FIBD

E. Clifford Dixon

E. Clifford Dixon, Major, USAF
Deputy Chief, WL/FIB

IF YOUR ADDRESS HAS CHANGED, IF YOU WISH TO BE REMOVED FROM OUR MAILING LIST, OR IF THE ADDRESSEE IS NO LONGER EMPLOYED BY YOUR ORGANIZATION PLEASE NOTIFY WL/FIBD WRIGHT-PATTERSON AFB OH 45433-7562 TO HELP MAINTAIN A CURRENT MAILING LIST.

Do not return copies of this report unless contractual obligations or notice on a specific document requires its return.

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	February 1997	**Final Report 2 Dec 95 - 1 Dec 96**

4. TITLE AND SUBTITLE
Interface-Driven Multidisciplinary Design of Large-Scale Aircraft Structures

5. FUNDING NUMBERS
C: F33615-96-C-3206
PE: 65502F
PR: 3005
TA: 41
WU: 52

6. AUTHORS
David R. Oakley, Graham S. Rhodes, and Lonny B. Kruger

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Applied Research Associates, Inc.
811 Spring Forest Road, Suite 100
Raleigh, North Carolina 27609

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)
Flight Dynamics Directorate
Wright Laboratory
Air Force Materiel Command
Wright-Patterson AFB OH 45324-7562
POC: Dr Max Blair, WL/FIBD, 937-255-6434

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
WL-TR-97-3078

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMLITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report presents Phase I research to develop an Interface-Driven Design Manager (IDM) that greatly reduces the design cycle time for affordable composite aircraft. The IDM represents a first attempt to fully integrate powerful new interface element and 3-D interactive graphics technology into a single design environment to automate the assembly and analysis of multicomponent global-local models for faster, more accurate composite airframe design. These emerging technologies have the potential for making multidisciplinary design optimization of large-scale composite structures practical and for providing new levels of design automation that are currently not possible. The IDM provides a graphical environment for rapidly assembling global-local models, as well as other complex multicomponent airframe models, from pre-meshed 'stock' components stored in a relational database, without concern for mesh compatibility. The IDM enables the designer to automatically insert components or regions with a highly refined mesh into the coarse mesh of a global model using interface elements. This provides two substantial benefits: (1) detailed local models can be used without remeshing the entire structure thereby substantially reducing the associated engineering cost; and (2) higher accuracy can be achieved in critical regions without substantial increases in computational cost. Both of these benefits make it practical to use higher-fidelity models earlier in the design cycle so that primary structures which are truly optimized for the application of affordable composites are achieved.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Aircraft Structures, Interface Elements, Global-Local Analysis, Finite Elements, Multidisciplinary Design, Interactive Graphics, Virtual Design Environment	**50**
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	**SAR**

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI std. 239-18

Table of Contents

List of Figures

ABSTRACT

This report presents Phase I research to develop an Interface-Driven Design Manager (IDM) that greatly reduces the design cycle time for affordable composite aircraft. The IDM represents a first attempt to fully integrate powerful new interface element and 3-D interactive graphics technology into a single design environment to automate the assembly and analysis of multicomponent global-local models for faster, more accurate composite airframe design. These emerging technologies have the potential for making multidisciplinary design optimization of large-scale composite structures practical and for providing new levels of design automation that are currently not possible. The IDM provides a graphical environment for rapidly assembling global-local models, as well as other complex multicomponent airframe models, from pre-meshed 'stock' components stored in a relational database, without concern for mesh compatibility. The IDM enables the designer to automatically insert components or regions with a highly refined mesh into the coarse mesh of a global model using interface elements. This provides two substantial benefits: (1) detailed local models can be used without remeshing the entire structure thereby substantially reducing the associated engineering cost; and (2) higher accuracy can be achieved in critical regions without substantial increases in computational cost. Both of these benefits make it practical to use higher-fidelity models earlier in the design cycle so that primary structures which are truly optimized for the application of affordable composites are achieved.

1. INTRODUCTION

1.1 Background

Large-scale aircraft design is presently a highly complex, exhaustive process involving huge manpower and computational resources. A complete aircraft structure is multidisciplinary in nature and is manufactured from a hierarchy of many different components. These components are joined together to form sub-assemblies which are then joined together to form larger assemblies and then finally assembled into a complete airplane [Niu 1988]. Due to the inherent complexities, airframe design traditionally proceeds in three phases: conceptual, preliminary, and final. The conceptual phase is centered on the layout and geometry of the primary surfaces and supporting framework; the preliminary phase defines the airframe internal structure and sizes the associated components; and the final design phase encompasses design details for individual components including interconnections and fasteners.

There are several problems with the current airframe design process that represent major barriers to the application of affordable composites to primary airframe structures. Composites have long been recognized as a novel way to reduce weight; however, their use in primary airframe structures has not been fully realized in military flight systems. One major reason for this is the lack of high fidelity modeling early in the design process. For example, preliminary design models do not provide accurate predictions of the stresses in critical regions (e.g., around cutouts and bonded joints). Thus, preliminary design models generally result in primary structures that are often designed with load path characteristics that exceed the material and manufacturing limitations of affordable composite structures. A second major reason is the lack of a bi-directional design process that can readily adapt to downstream design modifications needed to enhance the application of composites. The present design process is characterized by a feed-forward paradigm, whereby the design proceeds sequentially from conceptual to preliminary to final with little interaction or feedback between each phase. Ultimately, this results in a degree of inflexibility that can allow problem areas to become entrenched, thereby significantly reducing the potential for widespread application of composites in the final design.

Although powerful technologies have been developed to independently address key aspects of the design process, a unified virtual design environment that strategically integrates these technologies so that they have a major impact on the design process has not been developed. Current high-end CAD packages (e.g., Pro/ENGINEER, CATIA, and Unigraphics) have sophisticated database driven technology for building, viewing, and changing solid models of complex multicomponent systems. They also have sophisticated parametric technologies that allow full bi-directional associativity among components. Thus, a multicomponent system, such as a wing box, may be built up in such a way that if the planform of the wing changes, the location and geometry of the internal components (e.g., spars, ribs, etc.) is automatically changed as necessary. However, these systems generally do not work well for the complex geometries (e.g., blended regions) inherent in next generation aircraft structures. Also, these packages do not support critical emerging finite element modeling technologies such as interface elements that can greatly improve the speed and accuracy with which complex multicomponent models

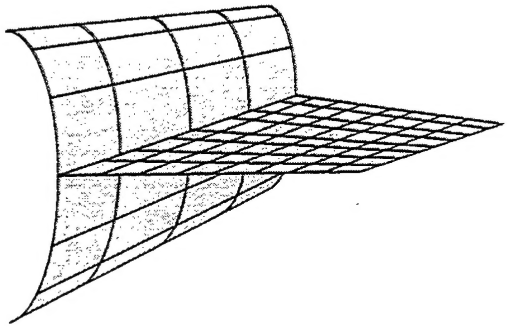
can be assembled and analyzed. Although limited analysis capabilities have been integrated into these CAD packages, seamless interfaces to sophisticated multidisciplinary analysis codes do not exist.

The present research is centered on developing an Interface-Driven Design Manager (IDM) that will greatly reduce the design cycle time for affordable composite aircraft. The IDM represents a first attempt to fully integrate powerful new interface element and 3-D interactive graphics technology into a single design environment to automate the assembly and analysis of multicomponent global-local models for faster, more accurate composite airframe design. These emerging technologies have the potential for making multidisciplinary design optimization of large-scale structures practical and for providing new levels of design automation that are currently not possible.

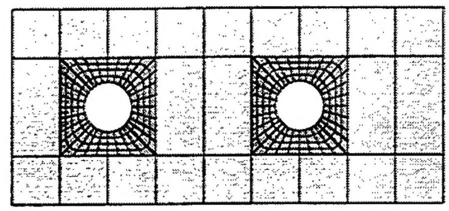
The IDM provides a graphical environment for rapidly assembling global-local models, as well as other complex multicomponent airframe models, from pre-meshed 'stock' components stored in a relational database, without concern for mesh compatibility. The IDM enables the designer to automatically insert components or regions with a highly refined mesh into the coarse mesh of a global model using interface elements. This provides two significant benefits: (1) detailed local models can be used without remeshing the entire structure thereby greatly reducing the associated engineering cost; and (2) higher accuracy can be achieved in critical regions without substantial increases in computational cost. Both of these benefits make it practical to use higher-fidelity models earlier in the design cycle so that primary structures which are truly optimized for the application of affordable composites are achieved.

1.2 Interface Elements

The interface element is a new type of finite element that joins structures with dissimilar meshes (see Figure 1.1a). Interface elements are particularly useful for global-local analysis, where they allow detailed local meshes to be created within the coarse meshes of global models, as illustrated in Figure 1.1b and Figure 1.2. In this way, interface elements may be used as an efficient means of obtaining more accurate stress results during preliminary design. The traditional need for transition modeling, which is often tedious and complicated, is eliminated, thereby substantially reducing the engineering cost. Moreover, the errors that typically arise from the introduction of distorted elements during transition modeling are also eliminated. Interface elements can also be used to assemble complex structural models from independently meshed components, without concern for mesh compatibility. For example, interface elements can join the internal structural components of a wing to the outer skin at arbitrary locations, independent of the structural mesh. This provides the potential to coordinate the mesh of the skin with the aerodynamic grid, thereby eliminating the need to interpolate aerodynamic loads onto the structural mesh.



(a) Dissimilar Mesh Example



(b) Global-Local Example

Figure 1.1 Example Interface Element Applications

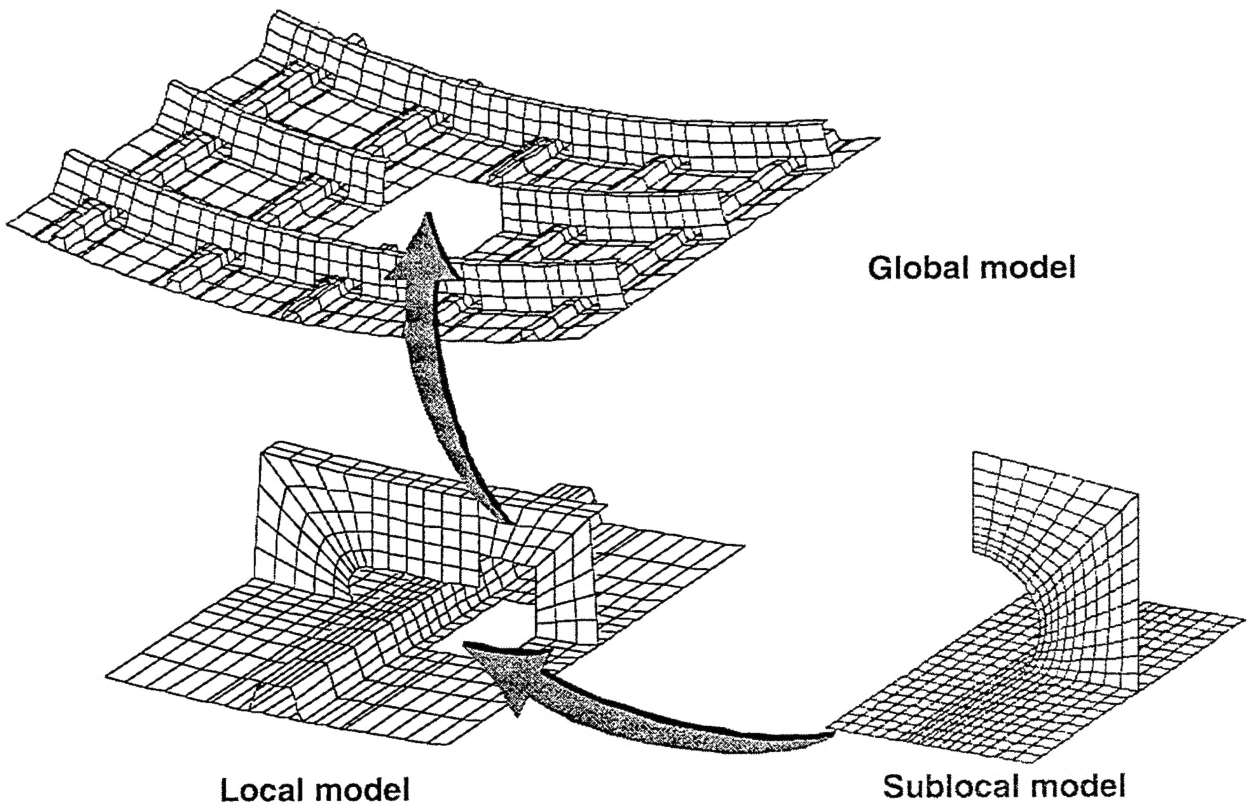


Figure 1.2 Nested Global-Local Modeling for Boeing Crown Panel

The one-dimensional interface element (a space curve) developed by Aminpour et al. [1992] is designed for joining independently modeled plate and shell substructures at their common boundaries. A pictorial representation of a 1-D interface element is shown in Figure 1.3, where three independently modeled substructures are being coupled together using a curved interface element. Formulated and cast in the form of a finite element, any number of interface elements may be used in an assembly of substructures. The interface element acts as a communicator between various substructures to accomplish proper load transfer and maintain deformation

compatibility. It has its own discretization (which need not match those of the connected substructures), and the deformation field over the interface element is described in terms of its nodal parameters. The interface element is based on a hybrid variational formulation that uses a constraint integral to enforce compatibility across the boundary between the substructures. In this way, all the desirable compatibility properties are achieved in a variational sense.

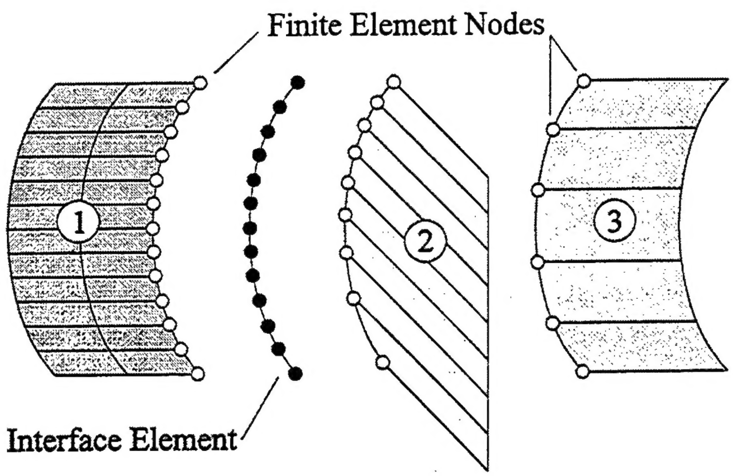


Figure 1.3 Illustration of One-Dimensional Interface Element

The 1-D interface elements described above have proven to be robust and accurate, in contrast to other interface approaches based on mortar elements [Maday et al. 1988] or spline fits [Shaeffer 1979]. More recent work [Davila and Aminpour, 1994] extends this technology to join shell structures across element faces. Work is underway at Applied Research Associates, Inc. to develop a 2-D interface element for NASA Langley Research Center, as shown in Figure 1.4.

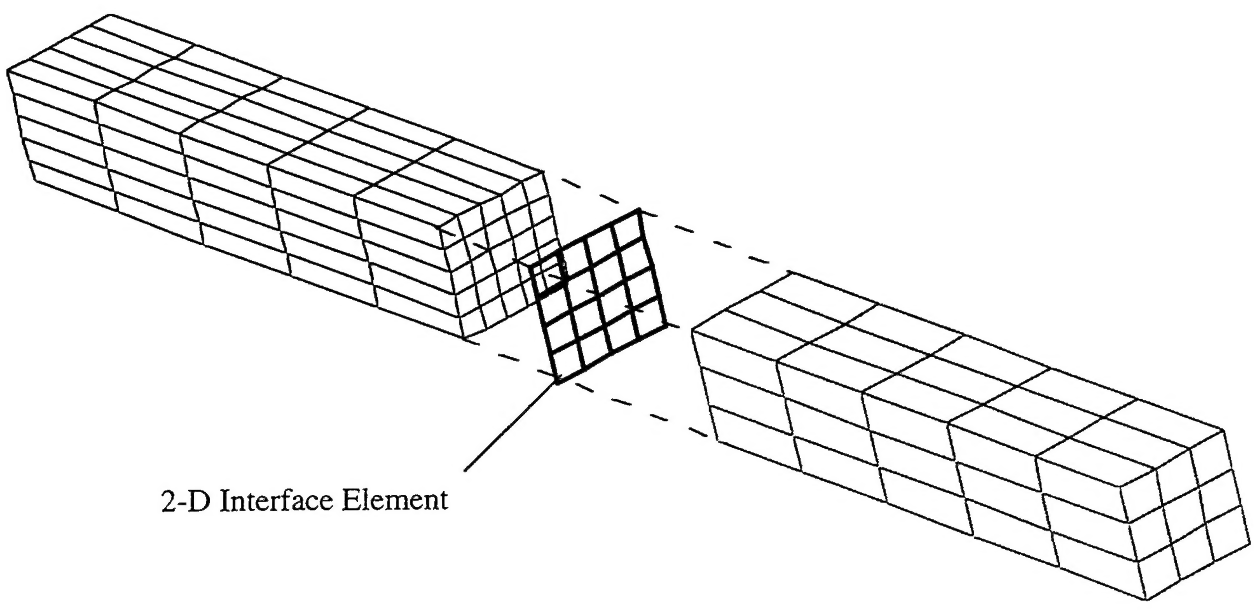


Figure 1.4 2-D Interface Element to Join 3-D Structures with Incompatible Meshes

1.3 Objectives and Scope

The overall goal of the Phase I effort is to develop and demonstrate a preliminary interface-driven design manager (IDM) for large-scale composite aircraft structures that will reduce the design cycle time and provide new levels of design automation that are currently not possible. The IDM integrates interface element technology into an interactive graphics environment to allow dissimilar meshes to be automatically coupled. This enables components or regions with a highly refined mesh to be automatically inserted into the coarse mesh of a global model to provide higher fidelity results during preliminary design, without large increases in engineering time. Interface elements also permit complex airframe structures, such as a wing box, to be rapidly assembled from a database of pre-meshed 'stock' components, without concern for mesh compatibility. The specific objectives for Phase I are to:

1. Develop a preliminary IDM design environment for rapidly assembling global-local models, as well as other complex multicomponent airframe models, from pre-meshed 'stock' components using interface elements.

2. Extend the capabilities of the preliminary IDM to enable the analysis of a multicomponent global-local model assembled using interface elements.

3. Solve two interface element example problems involving model assembly from stock components and global-local analysis to demonstrate the operation and features of the preliminary IDM system.

Under the first objective, we developed a baseline IDM with a 3-D interactive design environment that allows a designer to: (1) assemble airframe models from a database of pre-meshed 'stock' components; (2) insert refined components into coarse airframe models for global-local analysis; and (3) export assembled airframe models for finite element analysis. The graphical user interface for the baseline IDM consists of a main window, a stock components toolbar, and a three-dimensional modeling area. The 3-D interactive modeling environment allows the designer to select components or sub-assemblies from the database, drag these components into the modeling window, and 'snap' them together to rapidly build up complex models. The baseline IDM features a six-level relational database for storing the stock components. The database stores position and orientation information, references to the component geometry and finite element mesh files, and connection information that governs how components may be joined together. The baseline IDM includes the capability to display finite element meshes for each component. It also includes dynamic meshing capability that automatically adapts the baseline mesh of a stock component to accommodate resizing operations performed interactively by the designer.

Under the second objective, we developed and integrated two translators into the IDM. The first translator enables the designer to generate and export a COMET-AR input file for the assembled

structure.[1] This translator consolidates the component meshes for an assembled global-local model by: (1) parsing through the component database; (2) identifying common edges between components; and (3) automatically defining the interface elements that are needed to join the meshes between components. After the input file has been exported, the designer can execute a stand-alone version of COMET-AR to predict the static response. The second translator enables the designer to import and view the static analysis results from within the IDM environment. We also developed the capability to define boundary conditions (e.g., nodal loads and displacements) from within the IDM by clicking on a given node and entering specified force and/or displacement components.

Under the third objective we solved two interface element example problems to demonstrate the operation and features of the IDM system. The first example problem demonstrates use of the IDM to assemble and analyze a coarsely meshed wing box sub-assembly model from stock components using interface element technology. The second example problem demonstrates the IDM capabilities for efficient global-local analysis using interface elements. The IDM component assembly tools were used to graphically insert new spar components with refined meshes and circular cutouts into the wing box sub-assembly model, replacing components with the original square cutouts. Both of these test cases were analyzed to evaluate the impact of global-local modeling. Using the IDM analysis utilities, boundary conditions were defined and consolidated finite element input files were generated. COMET-AR was then executed to predict the static response, and the stress and displacement results for each test case were displayed and compared from within the IDM environment.

The remainder of this report is organized as follows. The methodology and numerical implementation of the baseline IDM are described in Chapter 2. In Chapter 3, the operation and features of the IDM are demonstrated using two different example problems. Final conclusions and recommendations are given in Chapter 4.

---

[1] COMET-AR is a modular, extendible, multilevel software system for computational structural mechanics research that includes interface element technology for connecting independently modeled finite element substructures along their common interface [Stanley et al. 1993].

2. METHODOLOGY

This chapter documents the methodology and numerical implementation of the baseline Interface-Driven Design Manager (IDM). The chapter begins with an overview of the features and operation of the IDM system. Afterwards, the following principal components of the IDM methodology are described in detail: stock component database, finite element import filter, interactive component assembly, and mesh consolidation.

2.1 Overview

The prototype IDM was developed with a 3-D interactive design environment that allows a designer to: (1) assemble airframe models from a database of pre-meshed 'stock' components; (2) insert refined components into coarse airframe models for global-local analysis; and (3) export assembled airframe models for finite element analysis. The graphical user interface for the baseline IDM is shown in Figure 2.1. It consists of a main window, a three-dimensional modeling area, and a stock components toolbar. The main window is a large window which serves as a container for the rest of the application. The stock component bar and the 3-D modeling window are children of the main window, and are contained within it, as are the pull-down menus and standard tool buttons.

The 3-D modeling window provides a unified, design-friendly environment for assembling large complex airframe models. The modeling window integrates powerful visualization and manipulation capabilities for positioning, resizing, and 'snapping' together stock components on the screen in real time. It can be used to graphically translate and rotate individual components or groups of components in space using OpenInventor manipulators (see Figure 2.1).[1] It can also be used to display the finite element mesh for each component. The modeling window provides many standard geometry visualization features, including the ability to view the airframe model from any angle with zoom and panning capabilities.

To simplify the process of model building, the IDM includes a relational database for storing the finite element models for 'stock' aircraft components. The stock components toolbar shown in Figure 2.1 is the graphical interface to this database. The database stores the position and orientation information needed to interactively orient a given component in 3-D space. It also stores references to the component geometry and finite element mesh files as well as interface connection point data.[2] The IDM's 3-D interactive design environment permits the designer to

[1] OpenInventor is an object-oriented three-dimensional toolkit for interactive graphics programming, developed by Silicon Graphics.

[2] Interface connection points are a set of discrete reference points along the boundary of each component that govern how components may be snapped together (see Section 2.4 for a detailed explanation of connection points).

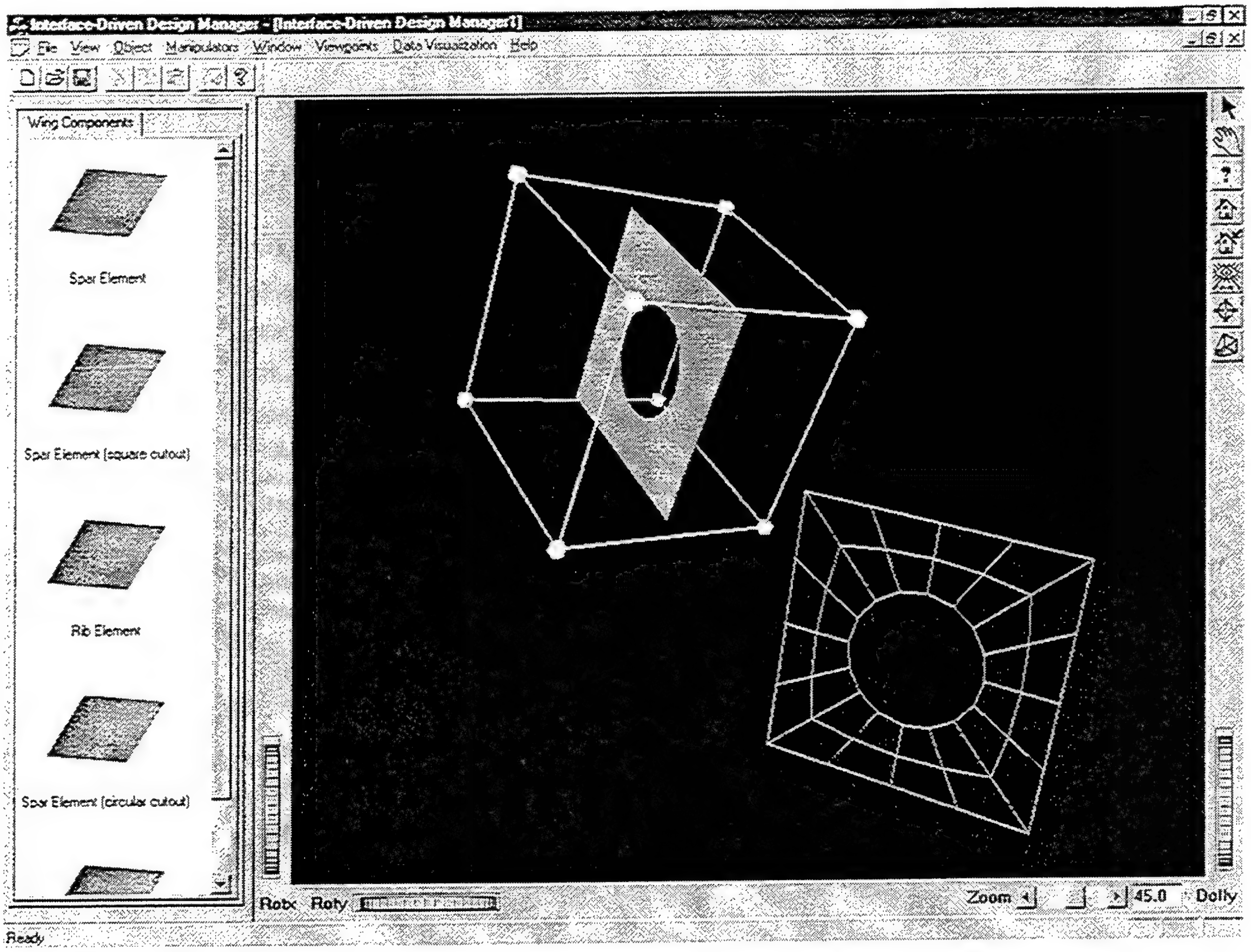


Figure 2.1 Graphical User Interface for the Baseline IDM

graphically select, customize, and 'snap' together stock components from this database to rapidly assemble an airframe structural model. The IDM uses interface element technology to join incompatible meshes between components in the global finite element model that it generates for the assembled structure. This further reduces the engineering time by freeing the designer to connect structural components together without concern for mesh compatibility.

Prior to using the IDM for the first time, the designer must populate the relational database with a core set of stock components. For a given component, this involves first creating a finite element model of the component and exporting this model in the form of a PATRAN neutral file. The next step is to identify points on the component which are useful points of alignment, that is, interface connection points at which the component will be connected or assembled together with other components (see Section 2.4). The final step in populating the stock component database is to specify categories for the different components - components are grouped by category for easy access when assembling a model. Currently, references to the PATRAN neutral file and connection point data must be manually entered into the database using a database management software package. However, once the database is created, the designer can exploit the inventory of stock components to rapidly produce many different new designs.

This database approach to model building is based on the fact that many airframe structural models can be assembled from a core set of 'building block' components (e.g., ribs, spars, etc.) that can be customized (i.e., resized and shaped) to accommodate a given design. Accordingly, the IDM database acts as a virtual warehouse of pre-meshed 'stock' aircraft components and sub-assemblies that can be used to rapidly build large-scale airframe models. Although new structural concepts will invariably require the use of some non-standard specialty components, once these components have been created, they can be added to the database to further expand the inventory of 'stock' components. A key factor in the success of this approach is the interface element technology, which permits pre-meshed components to be connected together without concern for mesh compatibility.

The IDM environment is designed to support rapid, user-friendly global-local analysis. Multiple versions of a given component can be stored in the database (i.e., a coarsely meshed version and a more detailed, finely meshed version as shown in Figure 2.2). For approximate response calculations, the fast-running coarse version of each critical component can be used. For applications requiring more detailed stress information, the refined version of each critical component can be automatically inserted into the coarse global model for global-local analysis. As before, interface elements are used to join the refined and global meshes along the boundaries. This provides two substantial benefits: (1) detailed local models can be used without remeshing the entire structure thereby substantially reducing the associated engineering cost; and (2) higher accuracy can be achieved in critical regions without substantial increases in computational cost. Both of these benefits make it practical to use higher-fidelity models earlier in the design cycle so that surfaces and primary structures which are truly optimized for the application of affordable composites are achieved.

(a) Coarse Version (b) Refined Version

Figure 2.2 Multiple Versions of a Structural Component

IDM Operation

To build an airframe model, the designer begins by graphically selecting several 'stock' components from the relational database. He then 'drags' each selected component over to the modeling window and 'drops' it into the desired position as shown in Figure 2.3. When the modeling window receives notification of the 'drop' event, it takes the information about the object, searches the database for the corresponding files, and then loads and displays the component in the window. Once a component has been placed in the modeling window, it can be dynamically positioned and resized using the IDM's powerful visualization and manipulation capabilities. The designer can graphically translate and rotate individual components or groups

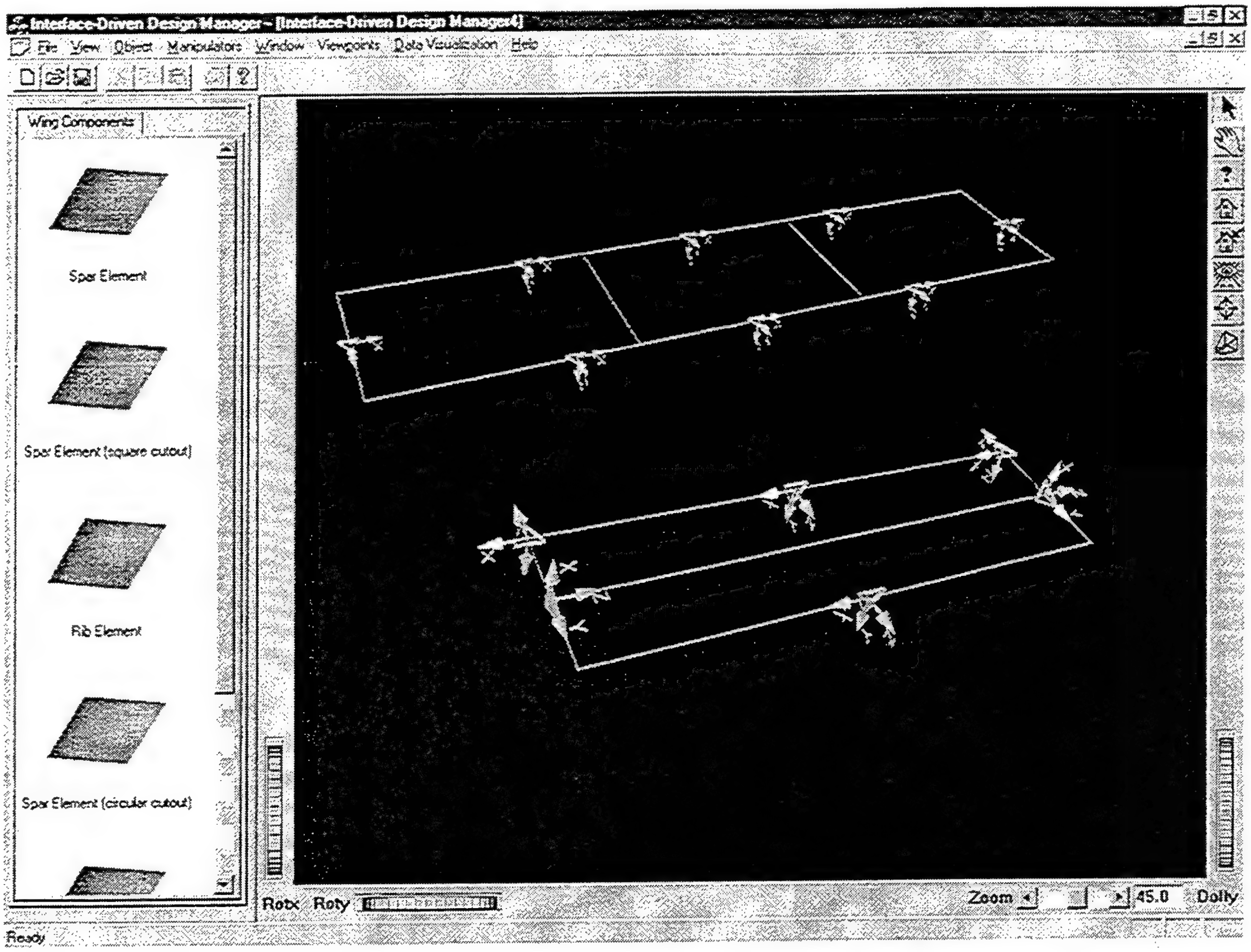


Figure 2.3 Stock Components Placed in the 3-D Modeling Window

of components in space using OpenInventor manipulators. In this way, components can be easily moved arbitrarily with respect to each other. The designer can also display the finite element mesh for each component. Furthermore, the IDM includes dynamic meshing capability that automatically adapts the baseline mesh of a stock component to accommodate resizing operations performed interactively by the designer.

After placing components in the modeling window, the designer can interactively 'snap' them together to build a multicomponent model (see Figure 2.4). This is done by graphically selecting two components to be assembled, identifying the source and destination connection points, and activating a button on the user interface. When this process is completed, the component with the source connection point translates and rotates in space to align with the destination connection point on the other component. Once again, the designer does not need to be concerned about mesh incompatibilities between components, since interface elements are used to account for this.

Once the structural model has been assembled, the designer can easily swap out individual

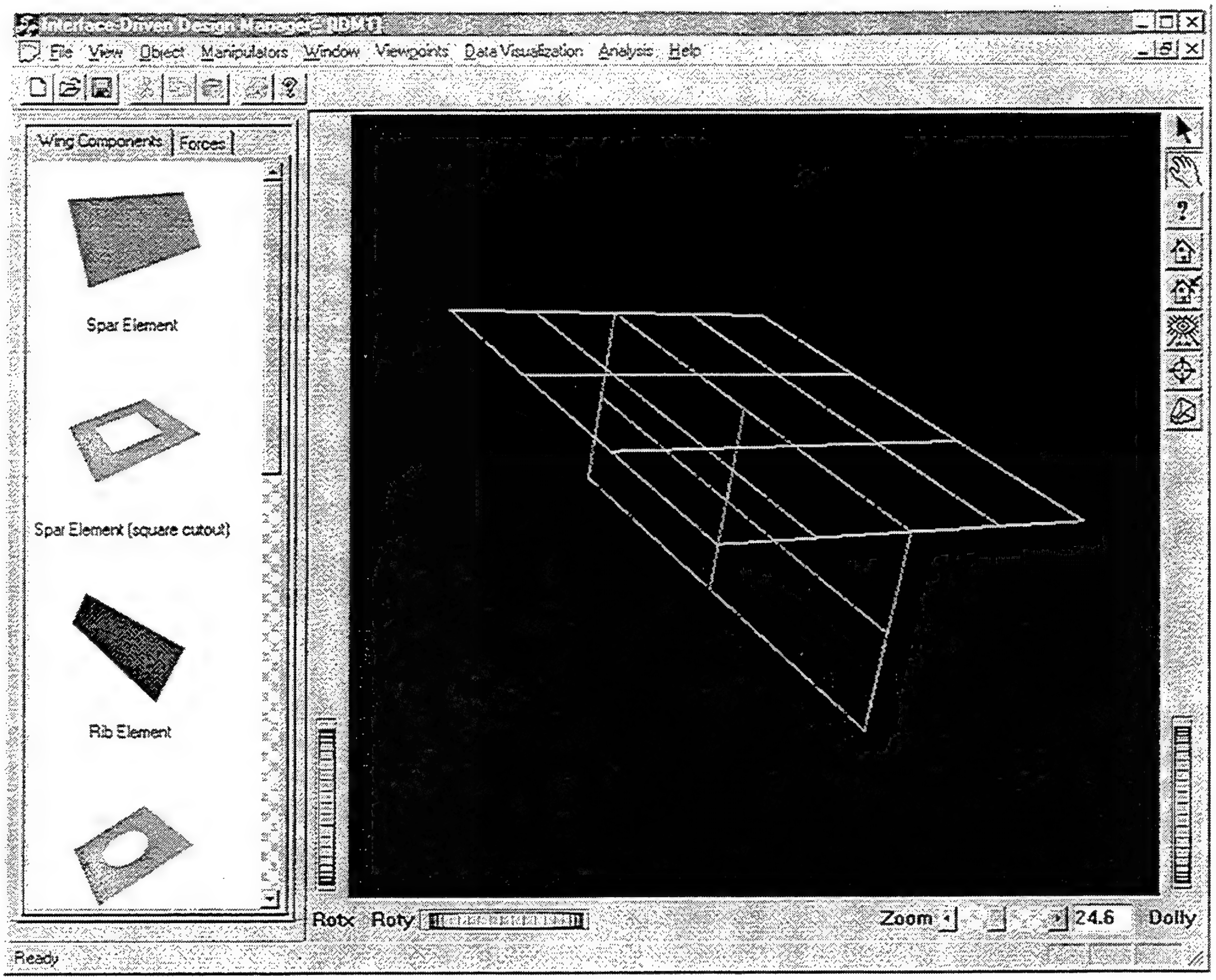


Figure 2.4 Assembling Stock Components into a Multicomponent Model

components to incorporate different levels of modeling detail. For example, the designer can delete a coarsely meshed component and insert a finely meshed version of the component to obtain more detailed stress information in critical regions (see Figure 2.5). Although not yet implemented, the IDM could also enable the designer to save the assembled model as a compound object, that is, an object comprised of two or more components. In this way, the database could be populated with hierarchies of compound objects in addition to the stock component objects. This functionality would be particularly useful for re-using previous models or sub-assemblies.

After the designer has assembled and saved a global-local airframe model, he can export it for independent finite element analysis using COMET-AR. The IDM environment enables the designer to define boundary conditions (e.g., nodal loads and displacements) by clicking on a given node and entering specified force and/or displacement components. Once the boundary conditions have been defined, the designer can analyze the structure by: (1) invoking a translator to generate and export a COMET-AR input file for the assembled structure, and (2) executing a

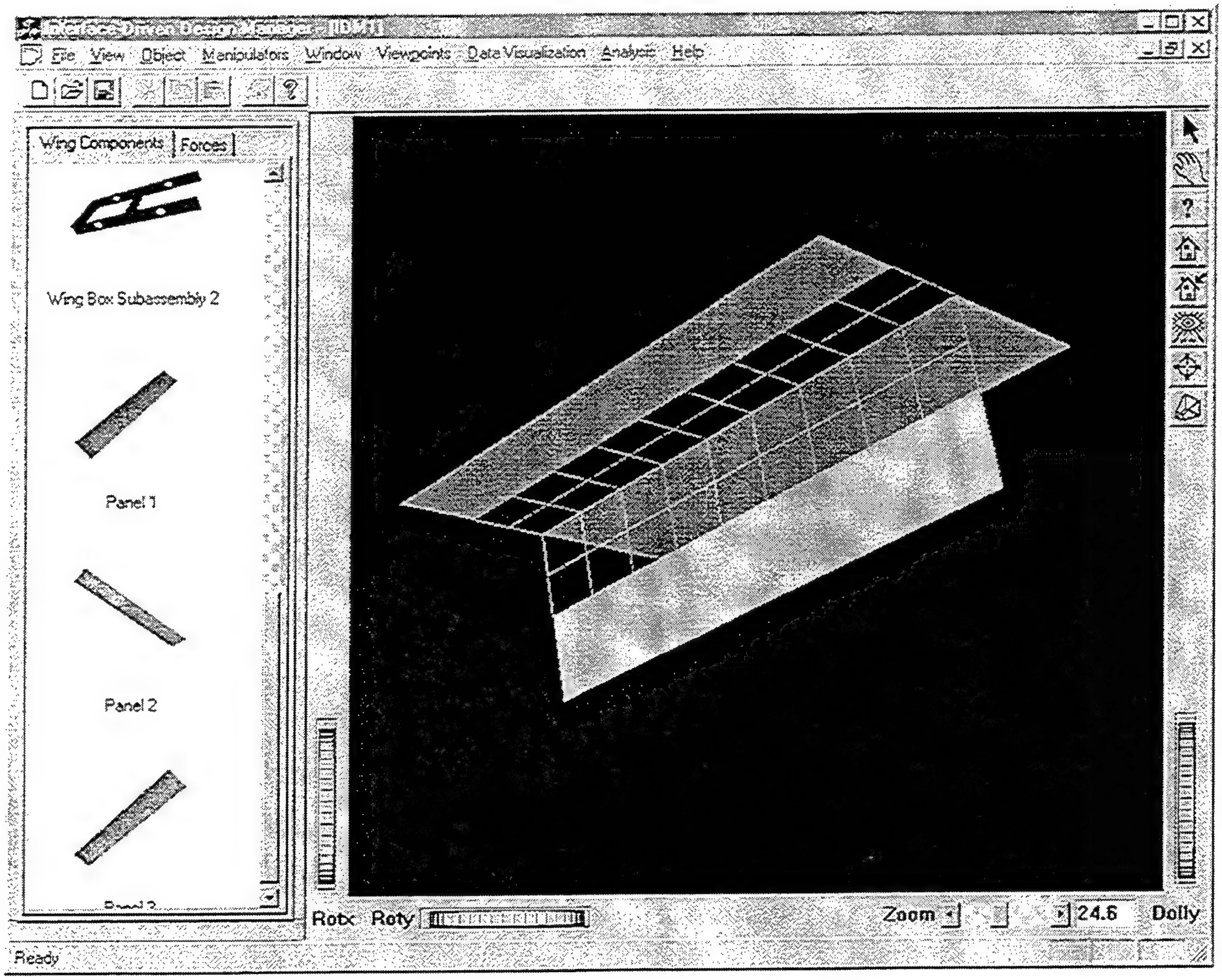


Figure 2.5 Inserting a Refined Component for Global Local Analysis

stand-alone version of COMET-AR to predict the static response. Using a second custom translator, the designer can import and view the static analysis results from within the IDM environment as shown in Figure 2.6.

2.2 Stock Component Database

The baseline IDM features a six-level relational database for storing 'stock' airframe components. The database stores the position and orientation information needed to interactively orient the components in 3-D space. It also stores references to the component geometry and finite element mesh files, and connection information that governs how components may be joined together. In subsequent developments this database may be extended to store assembled models (e.g., a global model with locally refined stock component inserts) as compound objects, that is, objects comprised of two or more components. The IDM would automatically define a unified set of attributes for the compound object based on the collection of attributes stored for

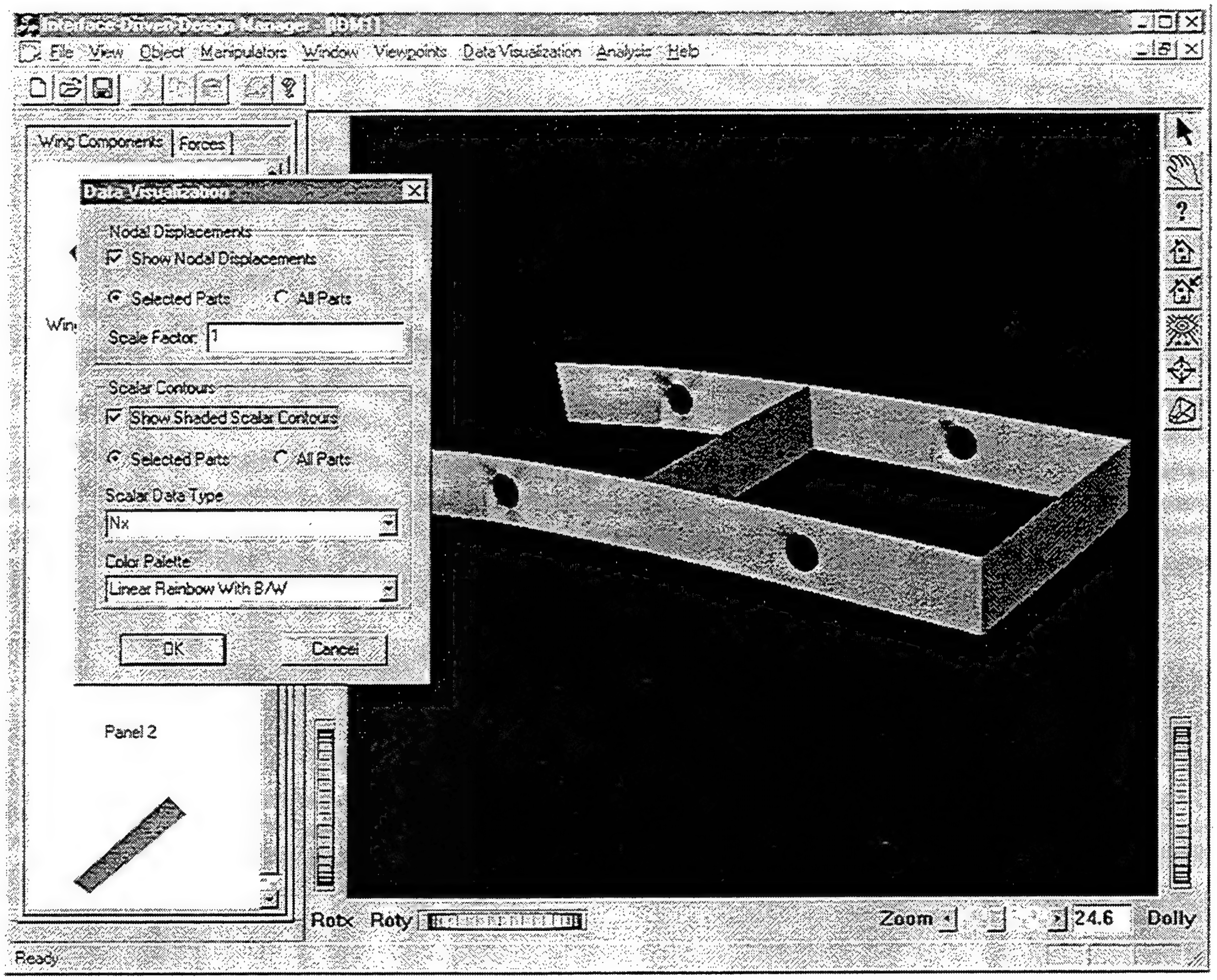


Figure 2.6 Displaying Analysis Results

its components. In this way, the database could be populated with hierarchies of compound objects in addition to the stock component objects. This functionality would be particularly useful for re-using previous models or model sub-assemblies.

The database schema is illustrated in Figure 2.7. The schema is designed to minimize table size, divide stock component objects into logical groups, avoid duplication of data, and speed access to the data. The first field in every table is the Primary Key. This field is an auto-incrementing number which assures that every record in every table is unique.

The top level table is the STOCKCATS table, a table of component categories such as "Aircraft Components" or "Automobile Components." The next table is called CATGROUPS, a table of subcategories owned by each component category such as "Wing Box Components" or "Actuators." The next table, CATGROUPITEMS, stores links associating individual stock components with the subcategories. The STOCKITEMS table contains data for the individual stock components, including a reference to the geometry and finite element mesh files for the component. The ITEMCONNECTPTS table contains all the valid connection points, separated

by type, for the stock components. As mentioned earlier, the connection points are a set of discrete reference points along the boundary of each component that govern how components may be snapped together. Each ITEMCONNECTPTS record contains a link to a stock component as well as position and orientation information in the local object coordinate system. This information is used to orient the object in 3-D space, when it is connected to another compatible component.

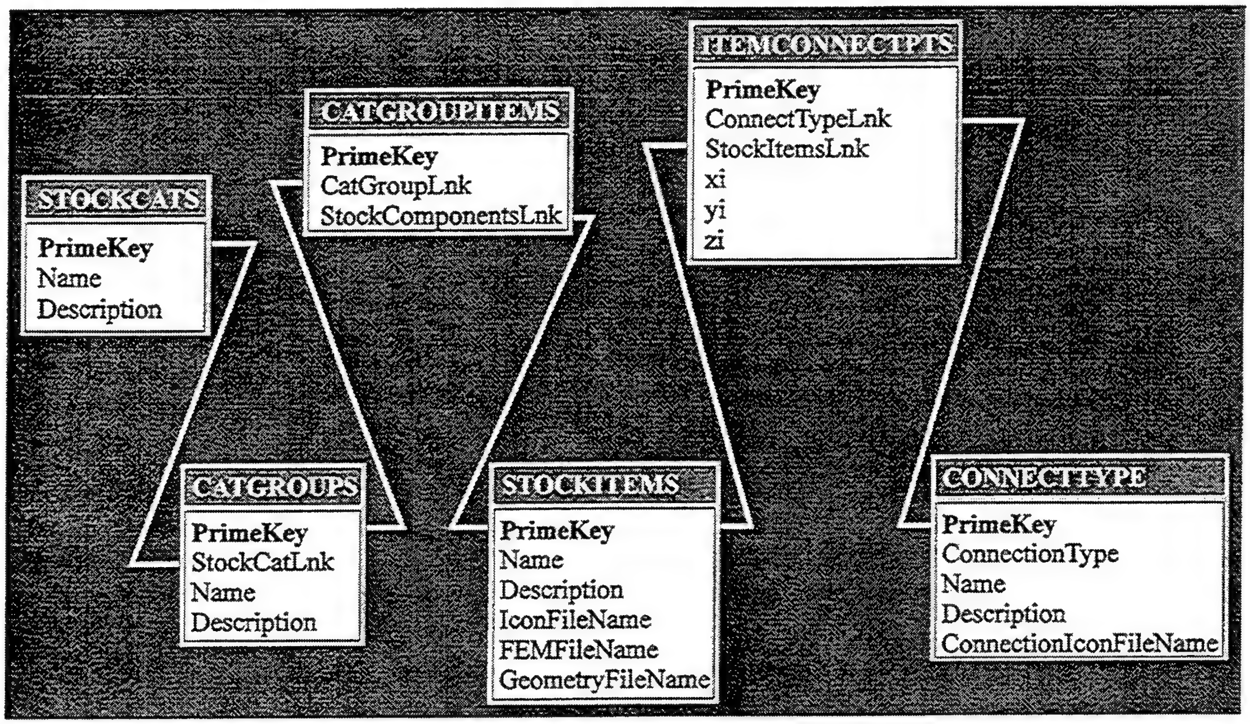


Figure 2.7 Stock Components Database Schema

The CONNECTTYPE table contains information on valid *connection type identifiers* for the stock components. A connection type identifier is assigned to each connection point, and is used by the IDM GUI to insure objects are connected only at compatible interface elements. For example, this information can be used to prevent the designer from connecting the edge of 2-D shell elements with the surface of 3-D brick elements using a 1-D interface element[1] (the technology to join 2-D shell elements to 3-D brick elements has not yet been developed). The connection type information can also be used to constrain assembly in a practical way. For example, wing box components may be assigned one connection type, and fuselage body components may be assigned another type. The system will then prevent the designer from inadvertently connecting fuselage body components to wing box components.

[1] The interface element between two 2-D meshes is a curve with only one independent coordinate and is called a 1-D interface element; the interface element between two 3-D meshes is a surface with two independent coordinates and is called a 2-D interface element.

We developed a stock component toolbar to serve as a graphical user interface to the database. The stock component bar is a persistent, dockable toolbar containing categorized links to the stock components. During an application, the IDM searches the database for information on the current layout of the stock component toolbar. It then populates the tabs of the window with the 'category names' for this configuration. Next, the program searches for the stock components which belong to these categories and places a bitmap representing the component on the corresponding tab in the stock component bar. The designer is then able to 'drag' the component over to the modeling window and 'drop' it into the desired position. When the modeling window receives notification of the 'drop' event, it takes the information about the object, searches the database for the corresponding files, and then loads and displays the component in the window.

2.3 Finite Element Import Filter

The IDM features an integrated PATRAN neutral file import filter that enables the designer to easily import finite element models for aircraft components into the relational database. The import filter was implemented as a set of cross-platform portable C++ classes, independent of any features specific to the operating system or OpenInventor. Furthermore, the import filter was written in a generic way and can easily be extended to accommodate alternative finite element formats (e.g., COMET-AR, NIKE3D, etc.).

The class hierarchy for the PATRAN neutral file import filter is illustrated in Figure 2.8. Here, the base-level class is *UnstructuredGrid*. The *UnstructuredGrid* class stores all information related to the mesh of a single stock component. This class is not derived directly from *GenericElement* and *GenericNode* (hence the dashed line in Figure 2.8). Rather, it includes data members which are lists of *GenericNode* and *GenericElement* objects. The most important aspect of the *UnstructuredGrid* class is that it provides the single interface for retrieving all finite element data, and for displaying and exporting a global mesh assembled with interface elements. Regardless of which finite element file is the source of the data, an object of type *UnstructuredGrid* is used to manage the mesh once it is loaded into memory.

The *GenericNode* class objects store all nodal data, including rectangular coordinates in three dimensions, displacements, velocities, stresses, etc. This class contains *method* functions for retrieving and setting nodal properties, effectively isolating the specific method of data storage from the code that needs to utilize the data. For example, a *method* function called *GetDisplacement()* is provided for retrieving the nodal displacement vector.

The *GenericElement* class objects store all element-specific data, including connectivity information and references to interface element data. This class contains *method* functions for retrieving properties of the element, such as element type. Additionally, functions are available for parsing the list of polygons that make up the element edge. These polygons reference *GenericNode* objects, which are used to generate an image of the element on the screen, including finite element results if requested by the user.

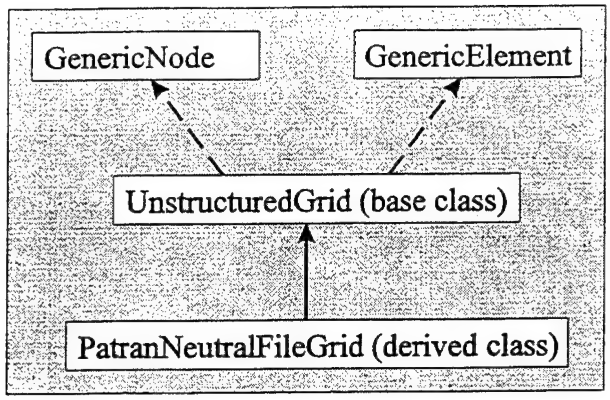


Figure 2.8 Finite Element Support C++ Class Hierarchy

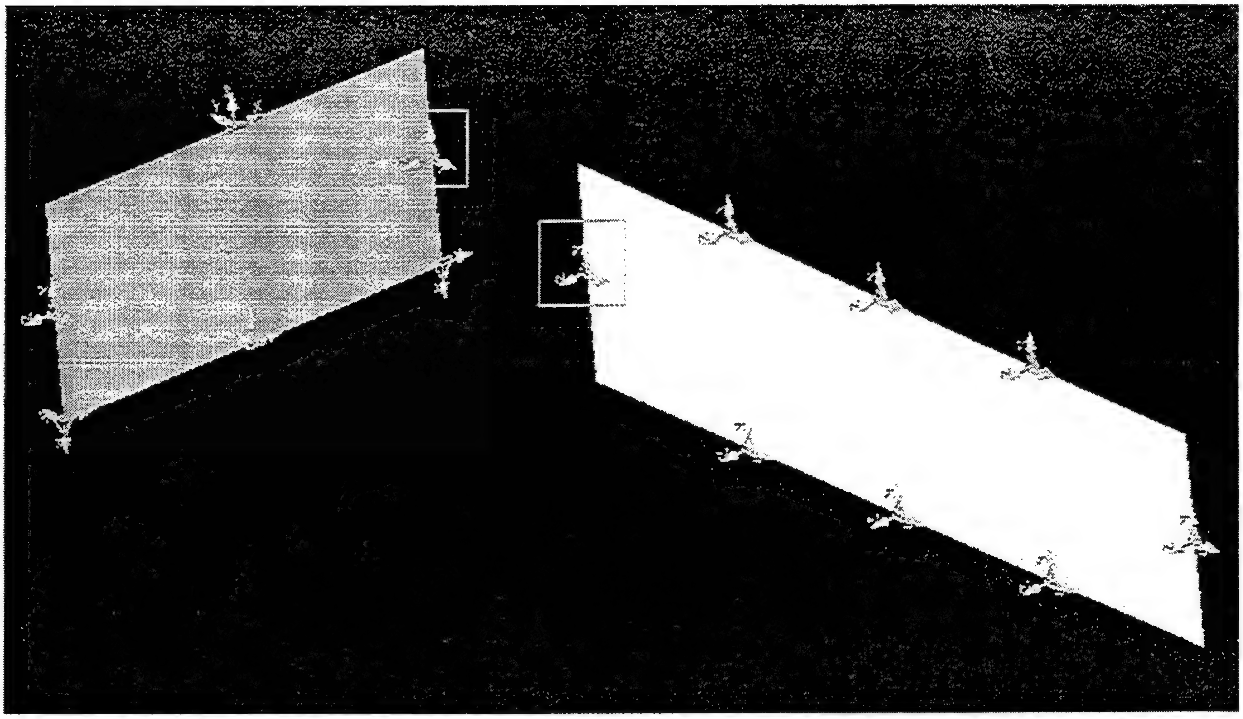
The last class is the *PatranNeutralFileGrid* class. This class is derived from *UnstructuredGrid*, and therefore contains all of the same functionality. Key methods of *PatranNeutralFileGrid* have been further developed to support PATRAN neutral files. When the stock component database contains a reference to a PATRAN neutral file, the IDM creates a new *PatranNeutralFileGrid* object. The IDM always references the object as an *UnstructuredGrid*; however, because PATRAN file support has been added, the object properly accommodates PATRAN neutral files.

2.4 Interactive Component Assembly

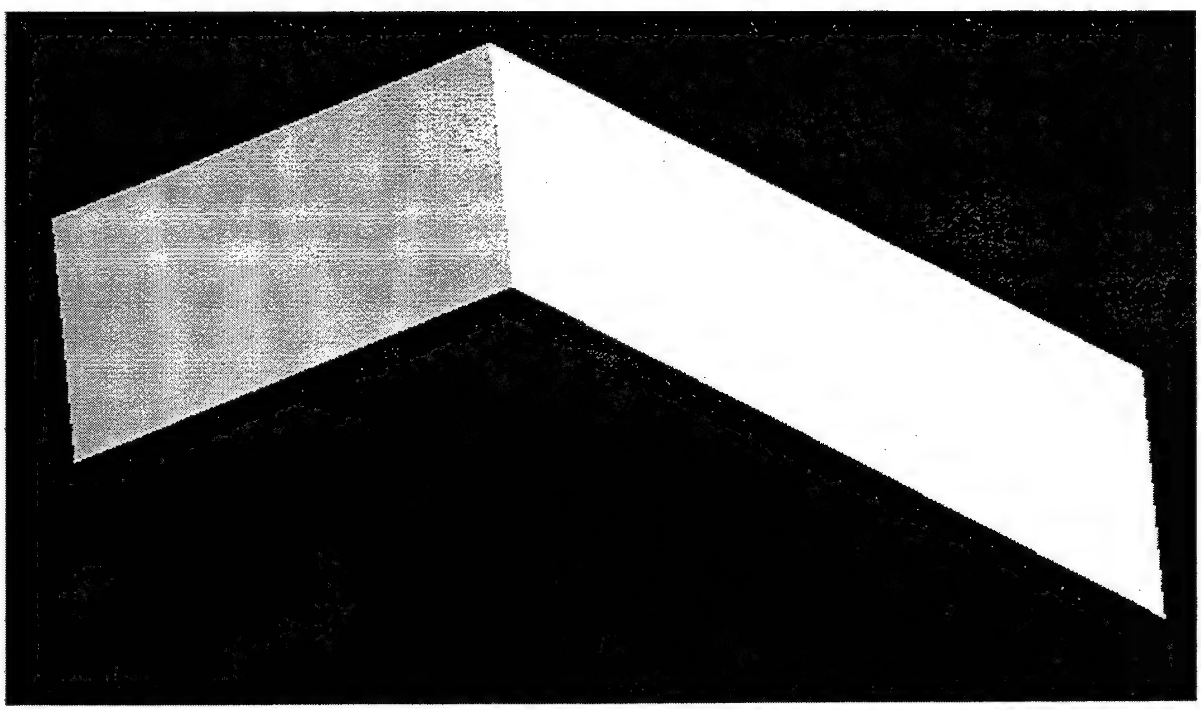
We developed the IDM design environment to support three-dimensional interactive assembly of stock components. The 3-D assembly capability is based on functionality that exists in the SGI OpenInventor toolkit using *Manipulators*. Manipulators are 3-D mouse activated controls that allow the designer to interactively change the position, rotation, and size of the components being used to build a global model.

A key part of our approach for interactive component assembly is the use of *interface connection points*. Figure 2.9 illustrates the concept of interface connection points. Each component has a set of connecting edges that are identified by the designer. These represent distinct nodal lines along the boundaries or surface of the component that can potentially interface with the connecting edge of another component. For each connecting edge in a component, the designer defines one or more *connection points* which act as distinct points of alignment for two components that are joined at this edge (see Figure 2.9a). Using these connection points, the process of component assembly occurs in two steps as follows. In step one, a connection point is selected on each component. In step two, the IDM automatically translates and rotates one component with respect to the other until the connection points are coincident and the connecting edges are colinear (see Figure 2.9b).

Thus, the connection points are model assembly tools that simplify the precise alignment of stock components. Connection points exist because, although the IDM provides manipulators which



(a) Connection Points



(b) Assembled Components

Figure 2.9 Interface Connection Points

intuitively move stock components in space, it is not a trivial matter to attach objects to one another. As mentioned previously, connection points are generally located at an interface edge, that is, an edge where two stock components will be joined via interface elements.[1] Each connection point also has a *type* identifier that is used to constrain assembly in a practical way. For example, wing box components may be assigned one connection type, and fuselage body components may be assigned another type. The IDM only allows components to be connected together at connection points with compatible types. In this way, it prevents the designer from inadvertently connecting fuselage body components to wing box components.

Three-dimensional interactive assembly of stock components is illustrated in Figures 2.10 and 2.11 for a global-local model. The image shows some simple stock parts for assembling a wing box sub-assembly model. Currently, assembly is done in two stages. During the first stage, stock components are dragged into the model scene from the stock component bar, and are translated and rotated in space using OpenInventor *manipulators*. In this way, objects can be moved arbitrarily with respect to each other. In Figure 2.10 for example, one component is highlighted with a *transform box* manipulator and may be moved with respect to the other two components.

During the second stage of assembly, objects are snapped together precisely at pre-defined interface connection points. Figure 2.11 illustrates partial assembly of the global-local model. The highlighted components have been assembled at a connection point, illustrated by a small rectangular coordinate system icon. The triangle shape at the origin of the connection point is used to visually identify the face or edge that represents the interface. Currently, precise object assembly is done by graphically selecting two objects to be assembled, identifying the source and destination connection points, and activating a button on the user interface. When this process is completed, the object owning the source connection point is translated and rotated in space to align with the destination connection point.

In the future, our approach for 3-D interactive assembly can be extended to support real-time snap-together assembly of multiple components into global models via interactive dragging. In this scenario, the designer will dynamically drag components and/or sub-assemblies together to assemble a structural model. When the distance between the interface connection point on a component being dragged is within a specified radius (called the *pick radius*) of a compatible connection point on another component, the manipulator will automatically align the components and snap them together. In this context, our use of single-point connection constraints has the added advantage that the proximity check required for interactive snapping at each mouse movement is not computationally expensive. Mathematically complex methods of interactive assembly, such as real-time surface-to-surface collision detection, are computationally expensive, and cannot be performed on low- to medium-end personal computers and workstations. Alternative fast methods of interactive assembly, such as snapping a point to a selected attachment surface do not enforce proper interface element edge alignment.

[1] The interface elements themselves do not depend on the connection points as they are generated independently via a topological analysis of the intersection of each pair of stock components.

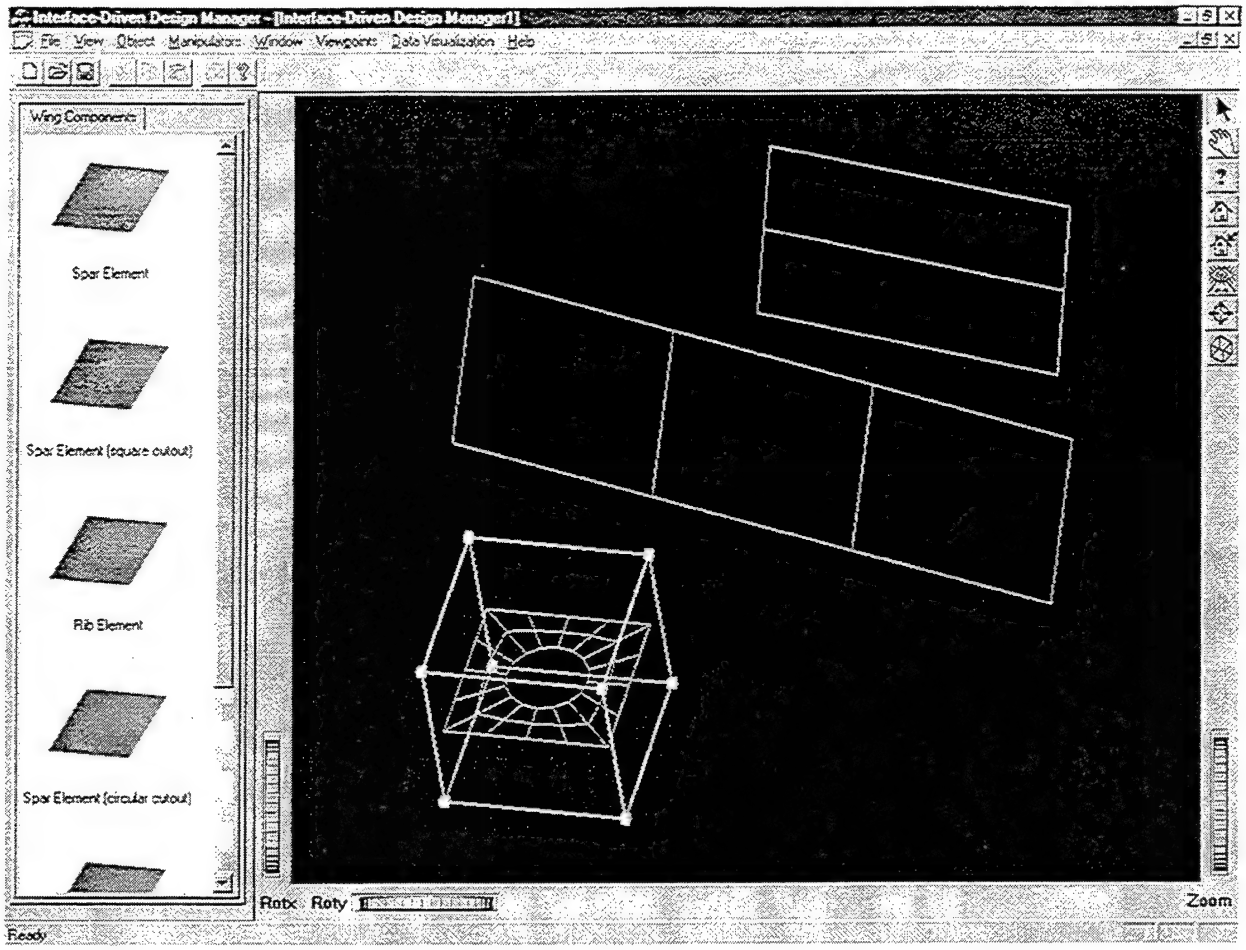


Figure 2.10 Stock Components for a Global-Local Model

2.5 Mesh Consolidation

We developed and implemented an algorithm that consolidates the component meshes for an assembled global-local model by: (1) parsing through the component database; (2) identifying common edges between components; and (3) automatically defining the interface elements that are needed to join the meshes between components. This capability is an essential IDM feature for building complex models from simple component objects. The IDM must be capable of generating a consolidated global mesh given two or more stock component meshes.

Figure 2.12 illustrates the consolidation of two component meshes, each consisting of a few shell elements. In this figure, all of the nodes of the two meshes are retained in the consolidated mesh. Node numbers and element numbers are simply shifted.[1] Whenever two stock components are

[1] Note that all part-to-part connections are implemented via interface elements, and so it is not possible to consolidate nodes between two components.

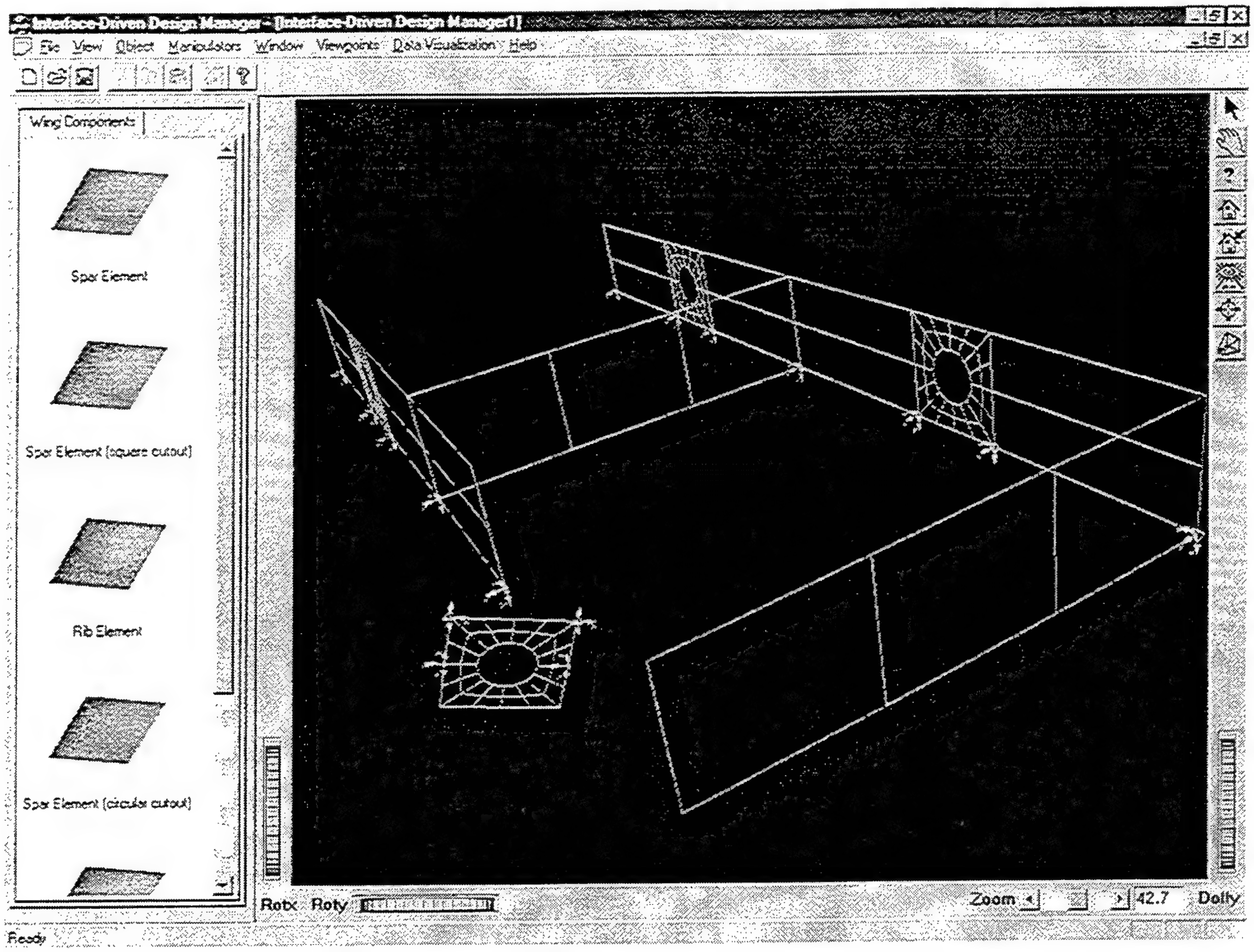


Figure 2.11 Interactive Assembly of the Global-Local Model

joined, the IDM parses through all of the elements in each mesh to identify those that have an edge in common. Each set of consecutive element edges that are shared between two meshes represents an interface between the two components. Afterwards, the IDM automatically generates a 1-D interface element for each distinct interface that is found.

It is not trivial to identify an interface when the stock meshes are not compatible at the adjoining edge, as illustrated in Figure 2.13. In this case, the IDM mesh consolidation algorithm performs a topological search to identify the nodes that lie on the interface edge of each mesh. Since the interface edges of each mesh are not exactly the same shape, the criteria for selecting nodes that represent each interface edge requires that: (1) elements along the interface edge of one mesh must be parallel (within some tolerance) to the corresponding elements of the other mesh; and (2) nodes along the interface edge of one mesh must be close (within some tolerance) to the interface edge of the other mesh. It is important to note that since the meshes of each component may be unstructured (consisting of an arbitrary sequence of nodes and connectivities) it is not known *a priori* which nodes lie on the interface edges of each mesh. Thus, all nodes including interior nodes must be considered, and those that are potentially part of an interface edge must satisfy both of the aforementioned conditions.

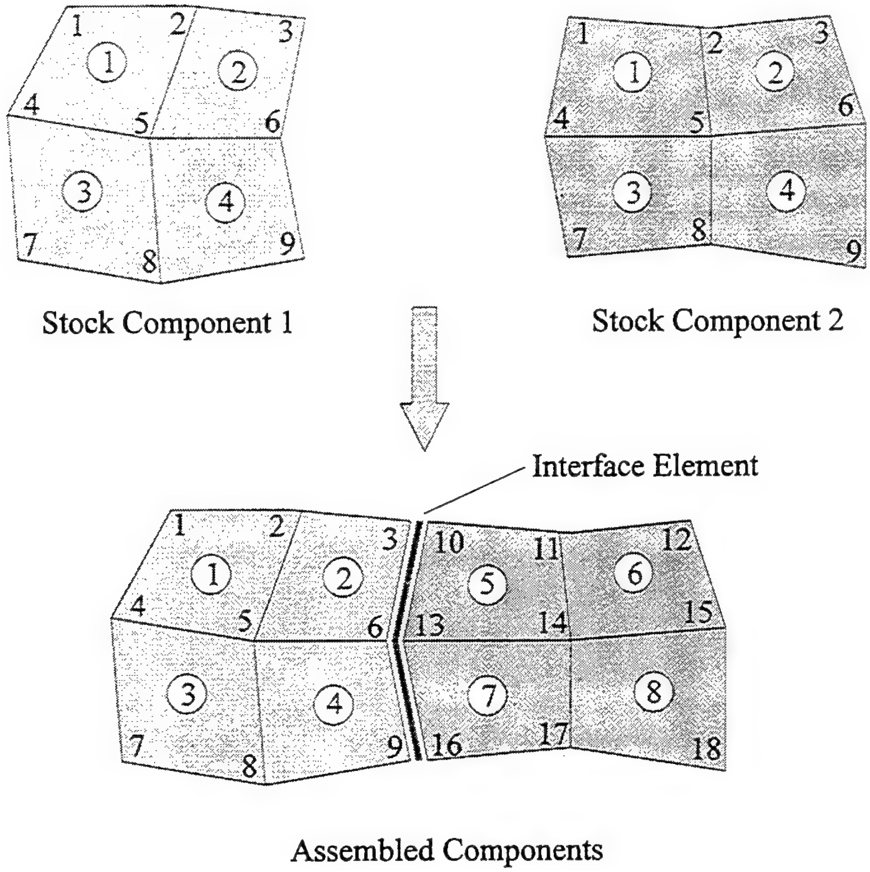


Figure 2.12 Consolidation of Two Stock Meshes

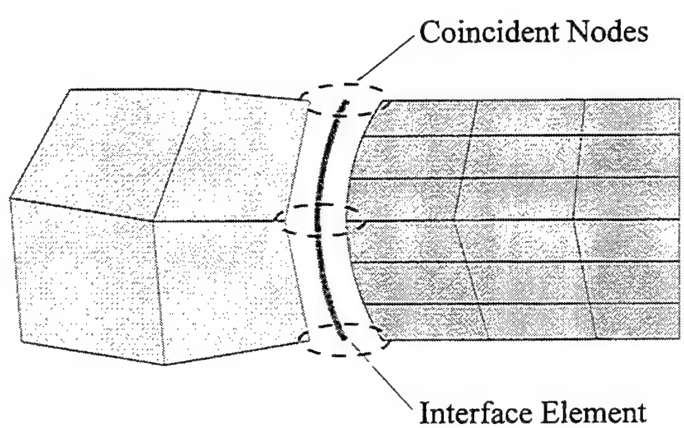


Figure 2.13 Meshes with Dissimilar Interface Edges

Once all of the nodes along an interface edge are known for both meshes, the interface element is generated. An interface element is defined by its own unique set of so called pseudo nodes. The set of pseudo nodes is known as the pseudo mesh. In some cases, the pseudo nodes are coincident with the real mesh nodes; however, in general they are not. To guarantee that the

interface element matrices are nonsingular, the number of pseudo nodes defining an interface element is constrained to be less than or equal to the number of nodes of the finest finite element mesh attached to the interface [Aminpour 1992].

It is important that the pseudo nodes generated for each interface element are consistent with the nodal spacing along the edge of each mesh that is being connected. To accomplish this, we generate *node spacing curves* for the interface edge of each mesh as shown in Figure 2.14. The vertical axis of the node spacing curve is simply an enumeration of node numbers along the interface edge. The horizontal axis is the normalized curvilinear arc length (bounded between 0 and 1) representing the distance along the curve of the interface edge for each node. As illustrated in Figure 2.14, the node spacing curve has a larger slope in regions where nodes are concentrated and a smaller slope in regions where nodes are sparse.

Node spacing curves are used to generate the pseudo nodes for an interface element as follows. The vertical axis of the node spacing curve is discretized into n - 1 uniform intervals, where n is the number of desired pseudo nodes. The corresponding horizontal axis values specify the distances of the corresponding pseudo nodes along the interface edge.

In cases were the interface element is shared by two or more interface edges, each with a different node spacing, it is desirable to generate a pseudo mesh with nodes consistent with all of the interface edges that are being connected. This is done by creating a *compound node spacing curve*, that is essentially a weighted average of all the node spacing curves from the shared interface edges (see Figure 2.15). The weighting factor assigned to each individual node spacing curve is equal to the number of nodes on the interface edge divided by the total number of nodes present on all of the shared interface edges.

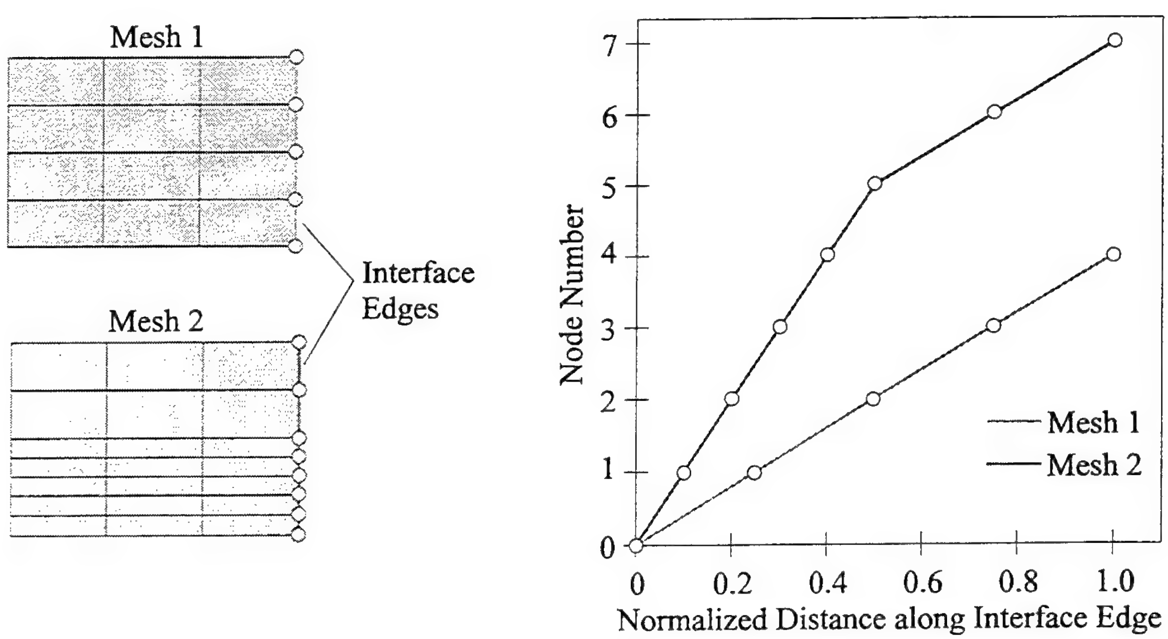


Figure 2.14 Node Spacing Curves for Interface Edges of Two Example Meshes

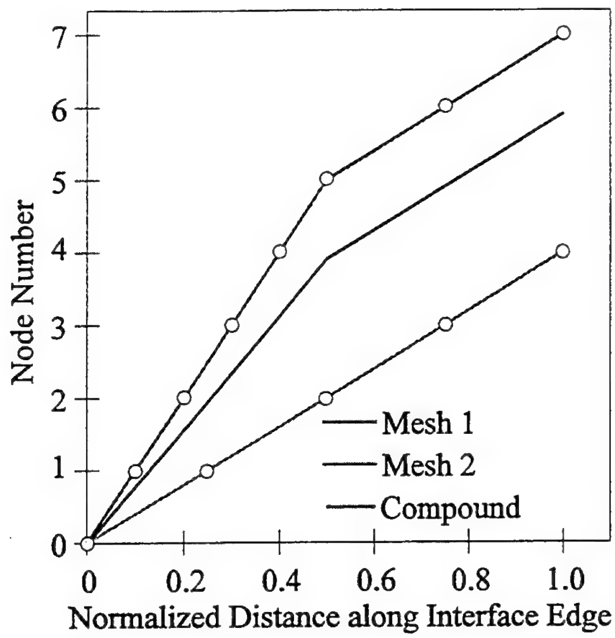


Figure 2.15 Compound Node Spacing Curve for Mesh 1 and 2 Interface

3. EXAMPLE PROBLEMS

In this chapter the operation and features of the baseline IDM are demonstrated using two different example problems. The first example problem demonstrates use of the IDM to assemble and analyze a coarsely meshed wing box sub-assembly model from stock components using interface element technology. The second example problem demonstrates use of the IDM 3-D design environment to rapidly create a global-local version of the first model using interface elements. Both of these test cases were analyzed to evaluate the impact of global-local modeling. Details regarding the assembly of each model are presented first, followed by a description of the analysis and results.

3.1 Example 1 - Coarse Wing Box Sub-Assembly Model

The first example problem demonstrates use of the IDM to assemble and analyze a multicomponent model from stock components using interface element technology. For this demonstration we assembled the wing box sub-assembly shown in Figure 3.1. Three different stock components were defined and stored in the component database: a solid spar component, a spar component with a square cutout, and a solid rib component (see Figure 3.2). Eight solid spar components, four spar components with cutouts, and two solid rib components were used to assemble the wing box sub-assembly.

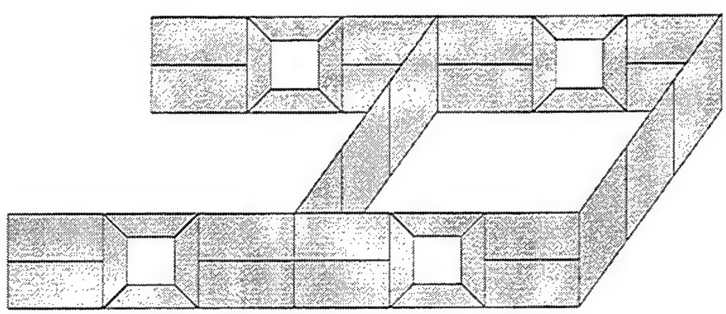


Figure 3.1 Wing Box Sub-Assembly for Example 1

Solid Spar Component | Spar Component with Cutout | Solid Rib Component

Figure 3.2 Stock Components used for Example 1

A diagram of the assembled finite element model of the wing box sub-assembly is shown in Figure 3.3. The outer dimensions are taken to be 60 inches in length, 24 inches in width, and 6

inches in height. The structural members are modeled using four-node assumed-natural-strain shell elements [Aminpour 1990] and a linear elastic material, and they are assumed to have a uniform constant thickness of 0.25 inches. The material of each member is taken to be aluminum with Young's modulus of 10.3 Mpsi and Poisson's ratio of 0.334. As shown in Figure 3.3, a total vertical load of 4000 pounds is uniformly distributed over the right end of the wing box model; the left end of the model is completely fixed. The coarse model of the wing box sub-assembly is discretized using 38 shell elements. In addition, 12 one-dimensional interface elements are required to assemble this model due to the meshing incompatibilities between components (see Figure 3.3).

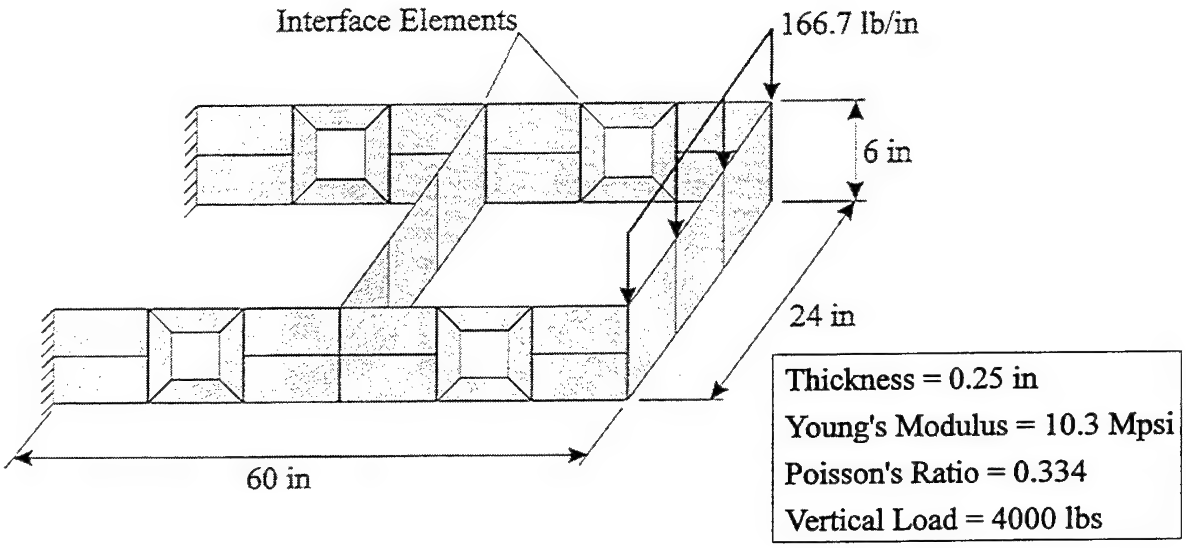


Figure 3.3 Finite Element Model for Wing Box Sub-Assembly

The baseline finite element model for each stock component was defined using PATRAN and then imported into the component database using the PATRAN import filter integrated in the IDM. The interface connection points for each component were defined using IDM utilities and stored in the component database. The interface connection points are shown in Figure 3.4 for each of the three stock components.

Using the 3-D interactive IDM modeling environment, multiple copies of each stock component were joined together to form the wing box sub-assembly (see Figure 3.5). After building the model, boundary conditions were defined by clicking on various nodes and entering the specified force and/or displacement components from a dialog box. This process is shown in Figure 3.6.

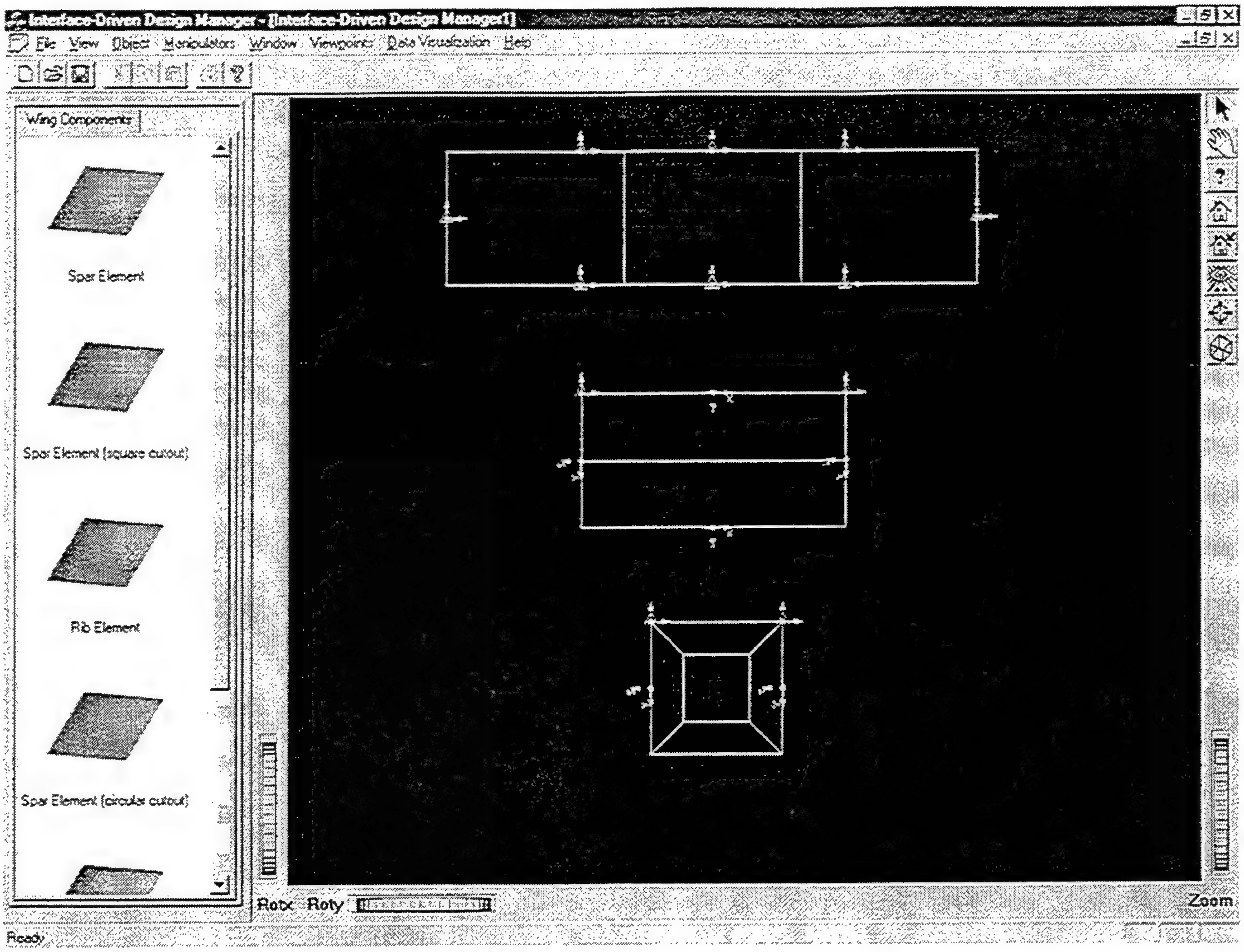


Figure 3.4 Interface Connection Points for the Stock Components

3.2 Example 2 - Global-Local Wing Box Sub-Assembly Model

The second example problem involves creating a global-local version of the first model, featuring locally refined spar components with meshes that more closely approximate the true circular geometry of the cutout (see Figure 3.7). This example demonstrates the IDM capabilities for rapid, user-friendly global-local analysis using interface elements. Specifically, the capabilities for creating and inserting a detailed or enhanced component into a global mesh and then analyzing the resulting hybrid (i.e., coarse and refined) model is demonstrated.

A refined version of the spar component with the square cutout used in the first example was created in PATRAN. As mentioned above, this version features a refined mesh that closely approximates the true circular geometry of the cutout and sufficiently predicts the local stress distribution. The refined spar component was imported into the component database using the integrated PATRAN import filter, and interface connection points were defined in the same manner as before.

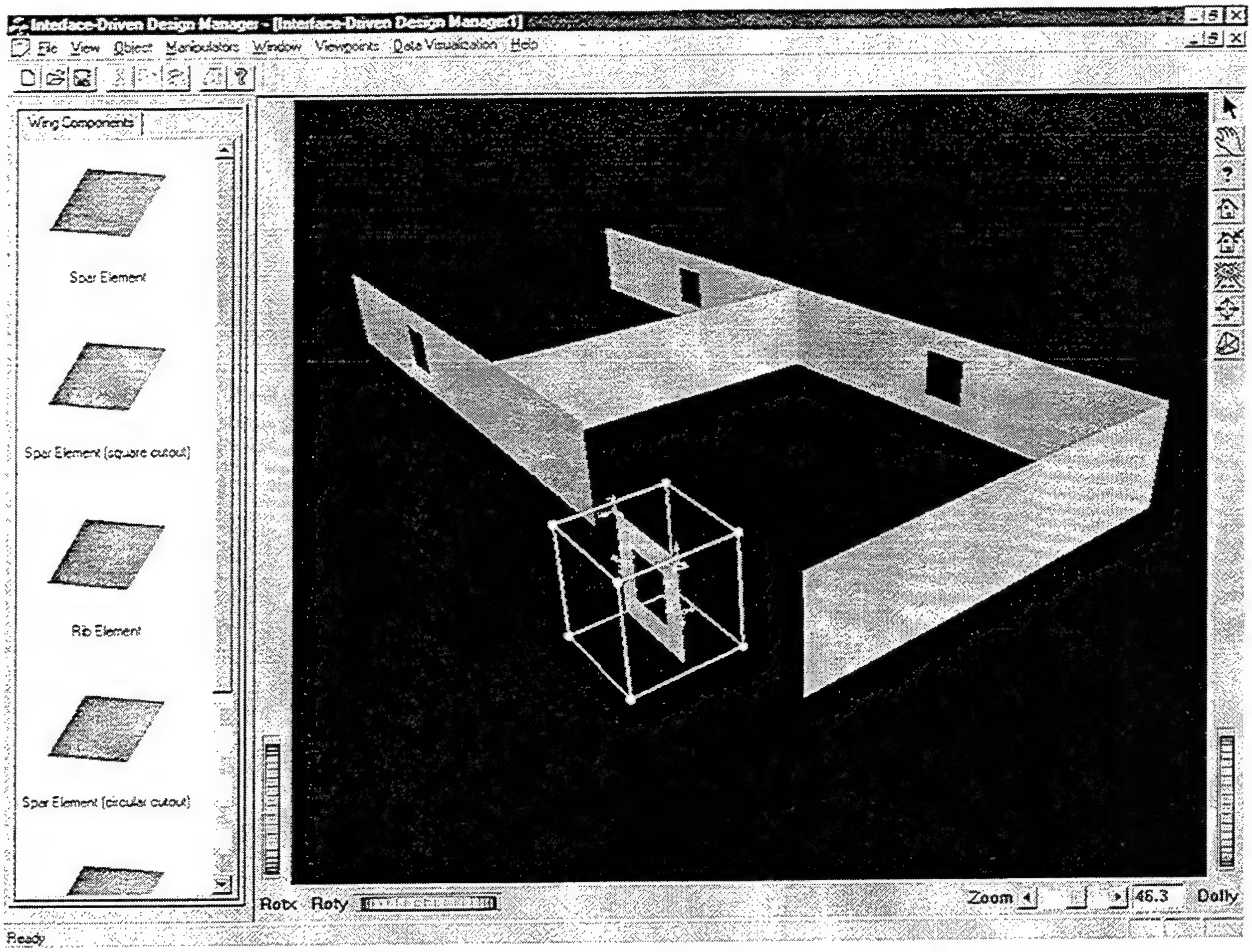


Figure 3.5. Interactive Assembly of the Initial Coarse Model

Using the graphical IDM component assembly tools, the new refined spar component was inserted into a copy of the wing box sub-assembly model, replacing components with the original square cutouts (see Figure 3.8). All of the other finite element information (e.g., boundary conditions and material properties) defined for the original model were retained for the new global-local model.

3.3 Analysis and Results

Both of the wing box sub-assembly models described above were analyzed (i.e., the initial model with coarsely meshed square cutouts and the global-local model with finely meshed circular cutouts). For each model, the Finite Element Model Export function was invoked to generate and export a COMET-AR input file for the assembled structure. A stand-alone version of COMET-AR was then executed to predict the static response. Using the Finite Element Results Import

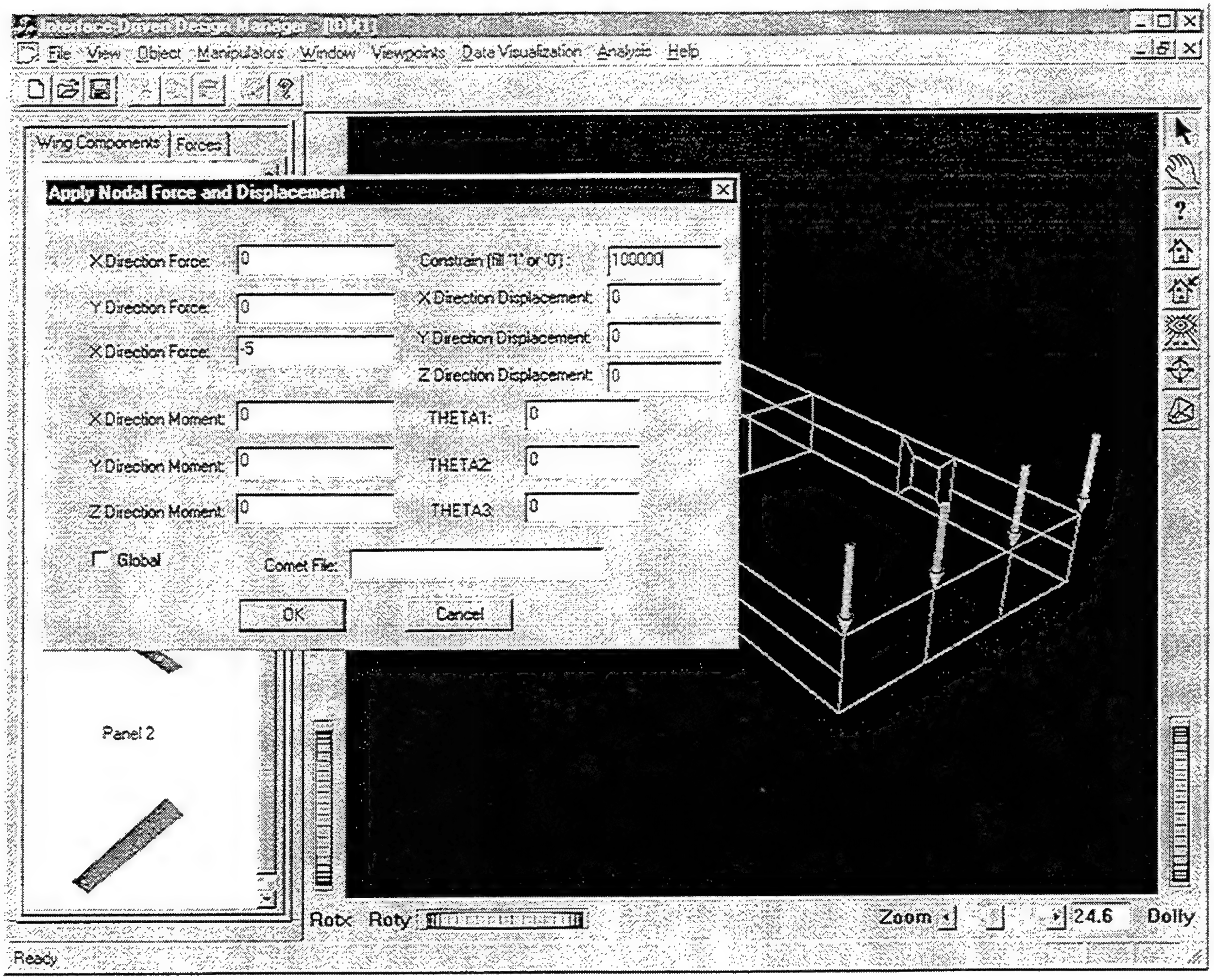


Figure 3.6 Definition of Boundary Conditions

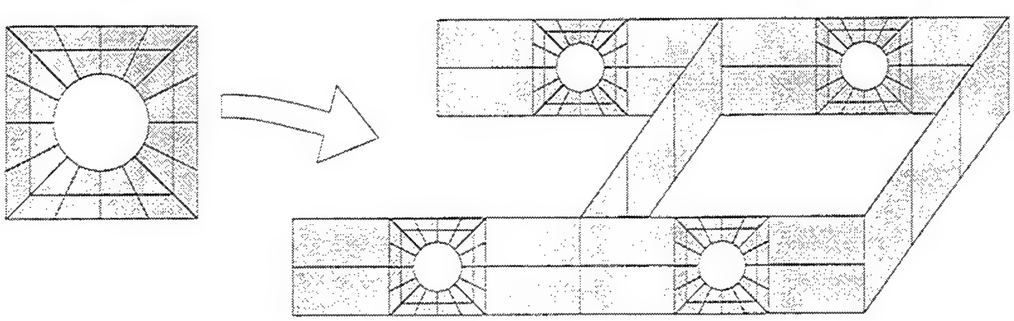


Figure 3.7 Global-Local Model with Refined Circular Cutouts

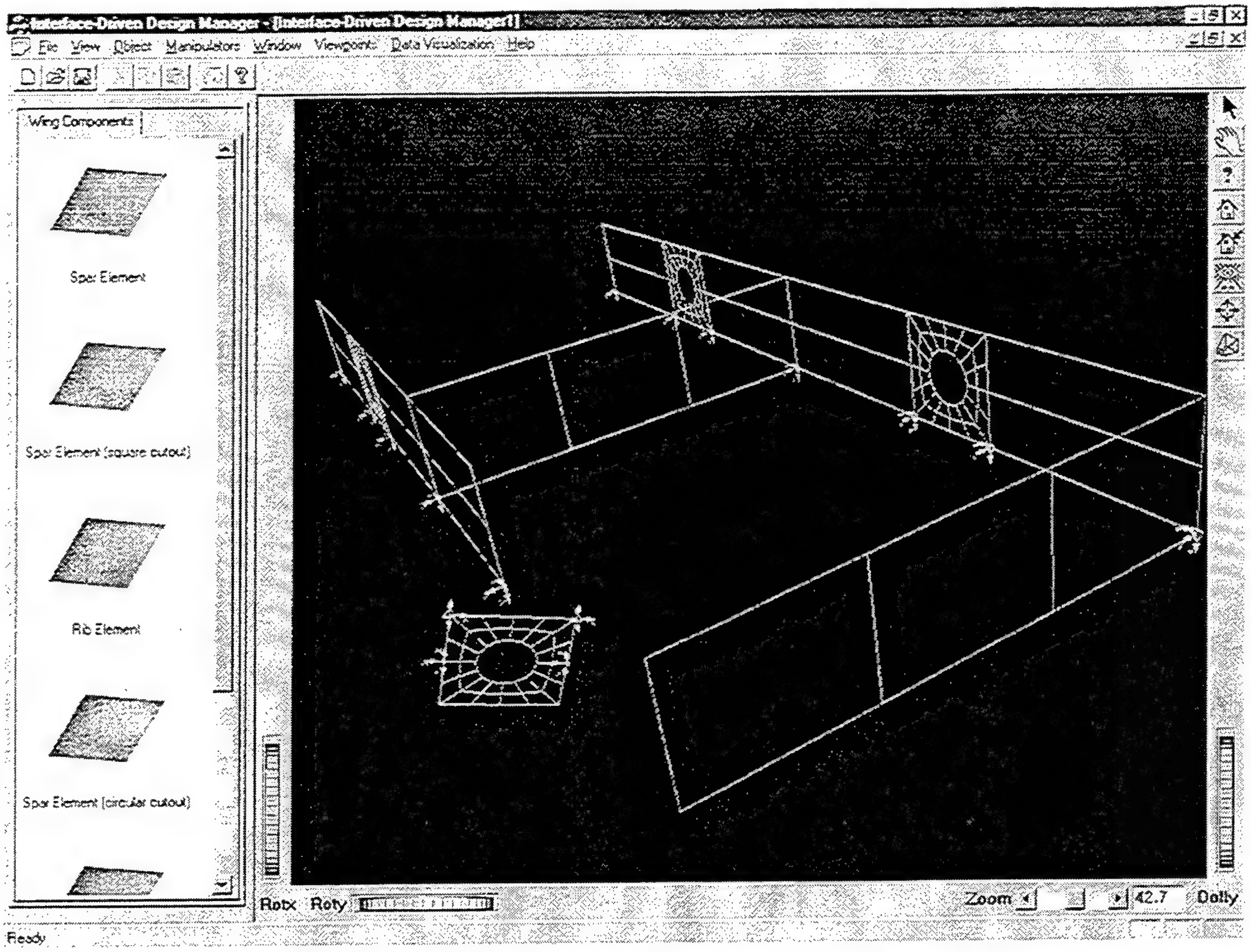


Figure 3.8 Interactive Assembly of Global-Local Model

function, the static analysis results were read from the COMET-AR output files and displayed from within the IDM environment.[1]

The deformed configuration of the global-local model is shown in Figure 3.9 (the deformed configuration of the coarse model is similar). As expected, the global-local model with the refined circular cutouts exhibits a slightly larger tip deflection than the coarse model (5.75 inches versus 5.68 inches). This is due to the additional compliance introduced by the locally refined portions of the mesh.

[1] The Finite Element Model Export and Results Import functions are built into the IDM interface and controlled by the analysis menu.

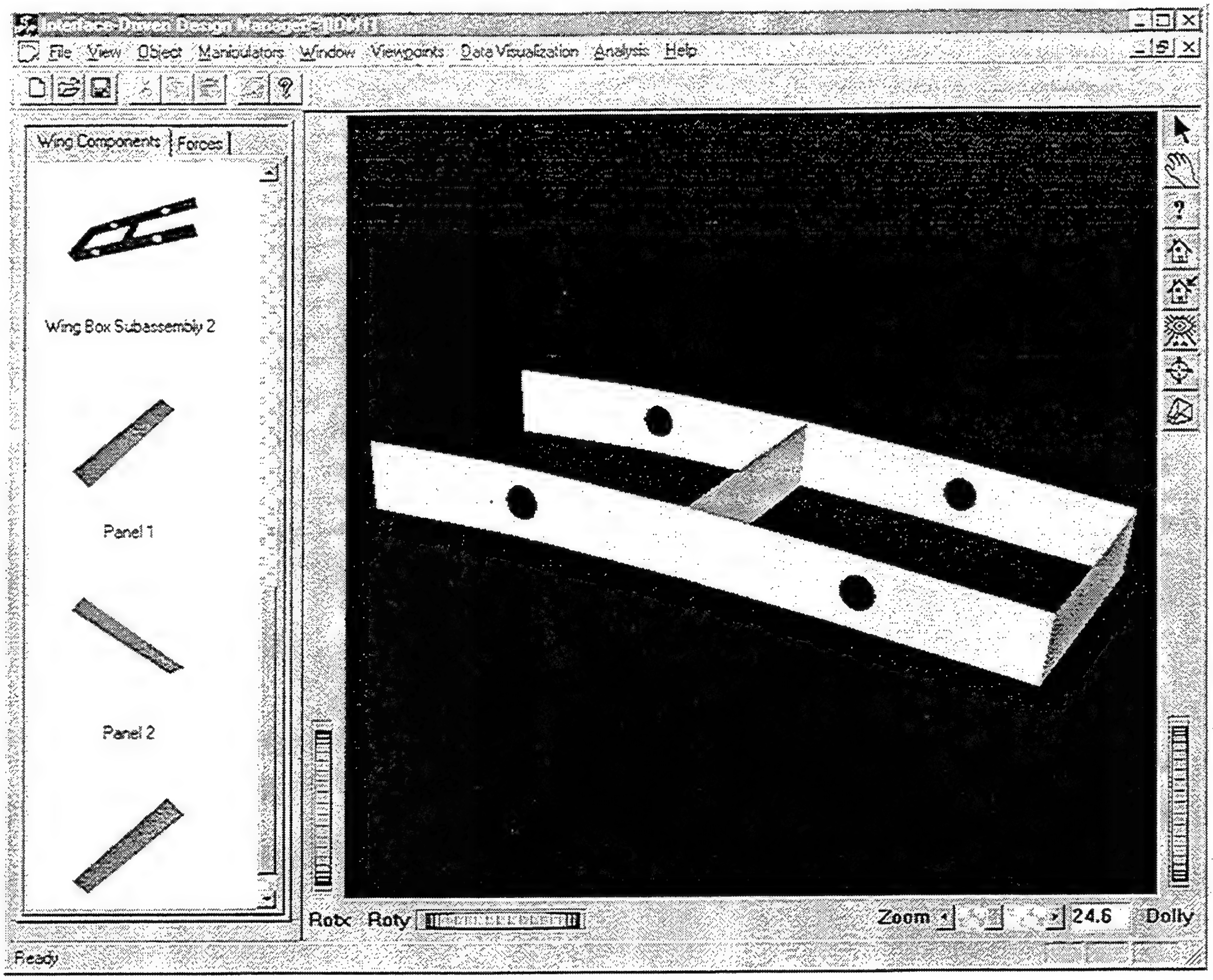


Figure 3.9 Deformed Configuration of the Global-Local Model

The stress contour results for each model are shown in Figures 3.10 and 3.11.[1] As expected, the global-local model provides a more detailed prediction of the stress distribution around the cutouts. The maximum stress around the square cutout is 32.2 Kpsi, whereas the maximum stress around the circular cutout is 96.3 Kpsi. This is significant, as it illustrates the impact that global-local analysis can have on analysis results and consequently on the design decisions

[1] It may be noted that the displayed stress contours in Figures 3.10 and 3.11 exhibit some discontinuities across the interface. This is a result of the use of different meshes on either side of the interface. Recall that the interface element enforces continuity of the displacements and tractions across the interface. However the stresses are computed in the postprocessing phase of the analysis, and the accuracy of the stress results is limited by the mesh discretization. The discontinuities observed in Figures 3.10 and 3.11 are due to a more accurate resolution of the stress resultants on the refined mesh compared to the coarse mesh.

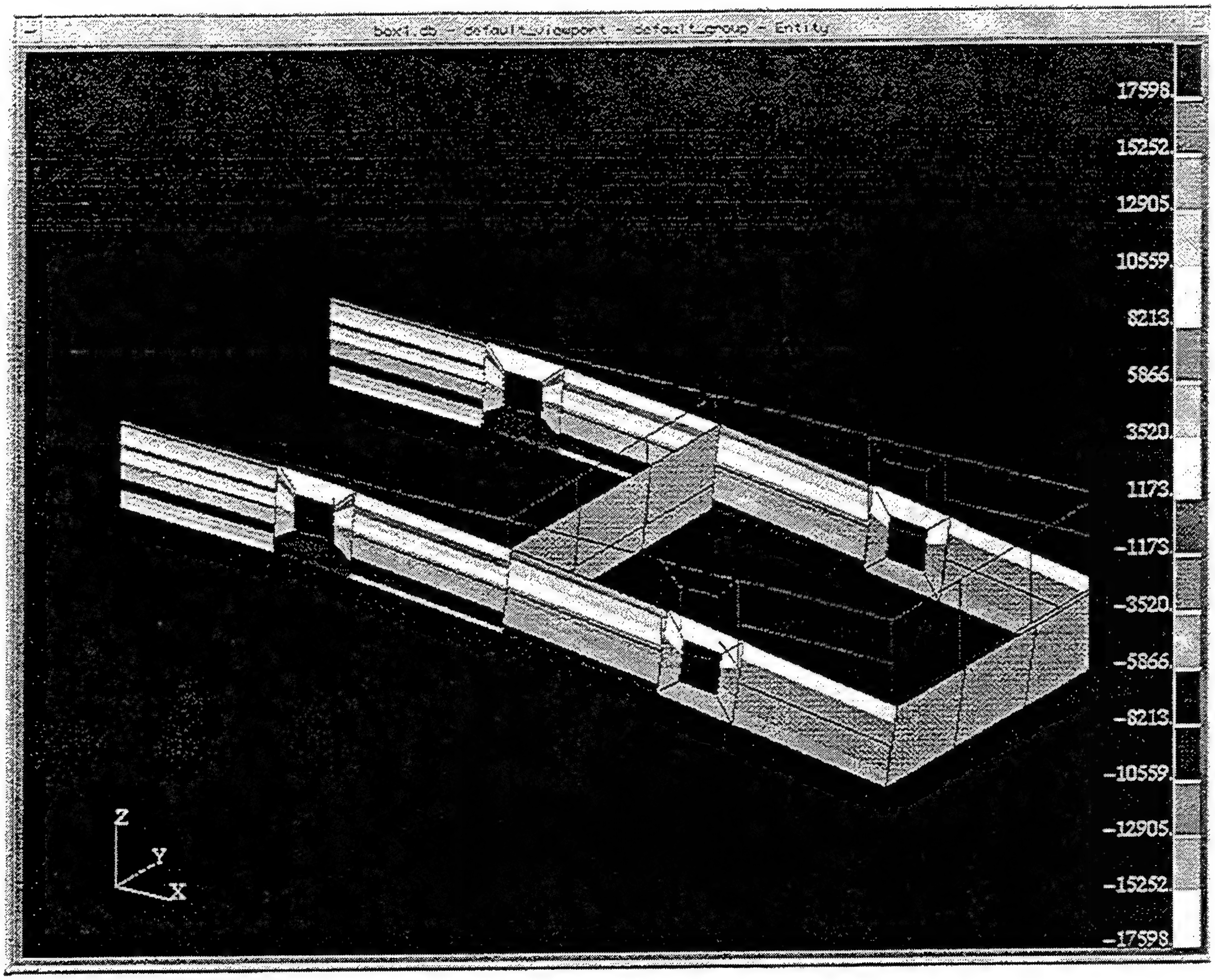


Figure 3.10 Stress Contour Results for Coarse Model

influenced by these results. Without accurate predictions of the stresses in critical regions, primary structures may be designed with load path characteristics that exceed the material and manufacturing limitations of affordable composite structures.

The baseline IDM developed herein takes a significant first step toward addressing this critical need. As demonstrated by these example problems, the IDM enables the designer to automatically insert components or regions with a highly refined mesh into the coarse mesh of a global model using interface elements. This provides two substantial benefits: (1) detailed local models can be used without remeshing the entire structure thereby substantially reducing the associated engineering cost; and (2) higher accuracy can be achieved in critical regions without substantial increases in computational cost. Both of these benefits make it practical to use higher-fidelity models earlier in the design cycle so that primary structures which are truly optimized for the application of affordable composites are achieved.

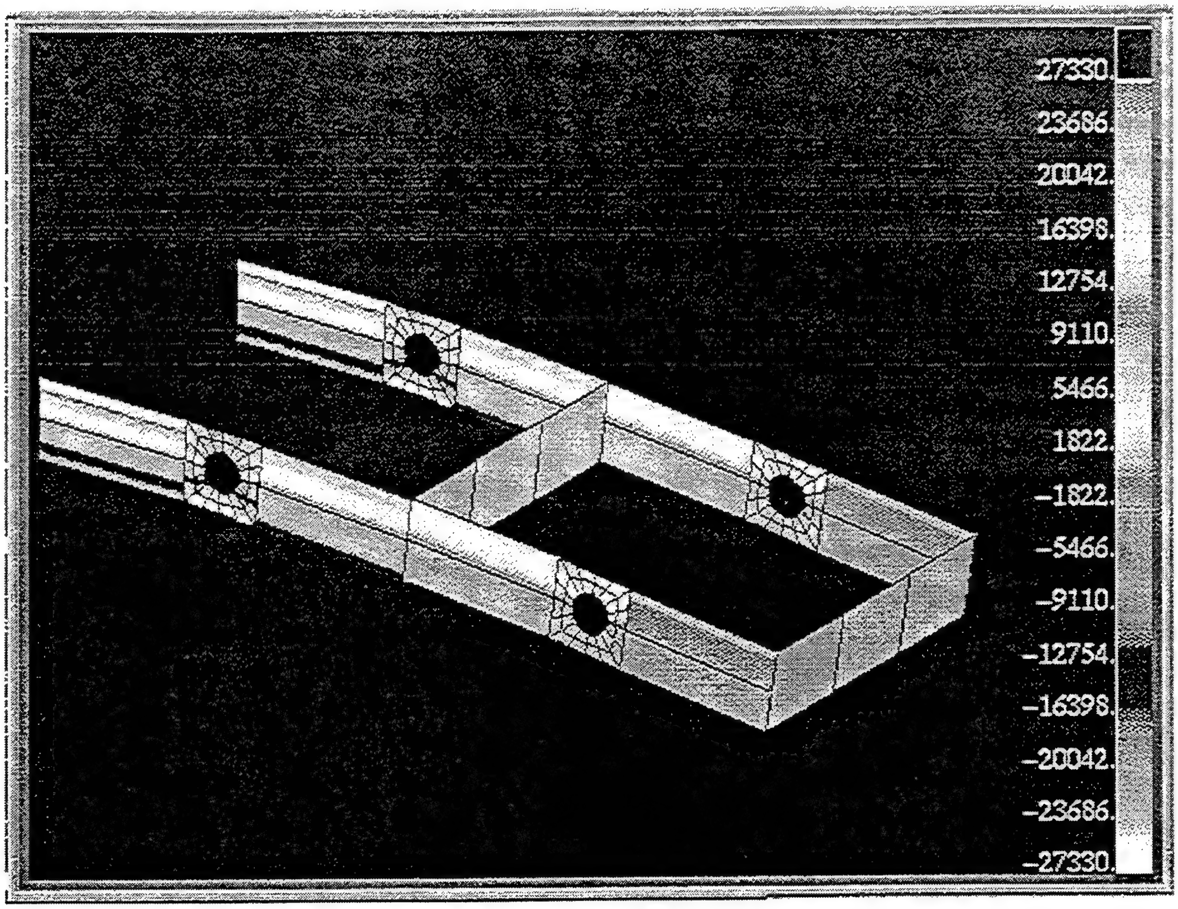


Figure 3.11 Stress Contour Results for Global-Local Model

4. SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

4.1 Summary

The overall goal of this Phase I research was to develop and demonstrate a preliminary interface-driven design manager (IDM) to reduce the design cycle time for large-scale composite aircraft structures. The IDM integrates interface element technology and a relational airframe component database into an interactive graphics environment to simplify the process of assembling complex structural models and performing global-local analysis.

We developed a baseline IDM with a 3-D interactive design environment that allows a designer to: (1) assemble airframe models from a database of pre-meshed 'stock' components; (2) insert refined components into coarse airframe models for global-local analysis; and (3) export assembled airframe models for finite element analysis. The graphical user interface for the baseline IDM allows the designer to select components or sub-assemblies from the database, drag these components into a 3-D modeling window, and 'snap' them together to rapidly assemble complex models. The baseline IDM includes the capability to display finite element meshes for each component. It also includes dynamic meshing capability that automatically adapts the baseline mesh of a stock component to accommodate resizing operations performed interactively by the designer.

We developed and integrated two translators into the IDM to support the analysis of global-local models assembled using interface elements. The first translator enables the designer to generate and export a COMET-AR model definition file for the assembled structure. This translator consolidates the finite element information for the assembled structure and automatically generates the interface elements that are needed to join the meshes between structural components. The second translator enables the designer to import and view the static analysis results from within the IDM environment. We also developed the capability to define boundary conditions (e.g., nodal loads and displacements) from within the IDM by clicking on a given node and entering specified force and/or displacement components.

We solved two interface element example problems to demonstrate the operation and features of the IDM system. The first example problem demonstrates use of the IDM to assemble and analyze a coarsely meshed wing box sub-assembly model from stock components using interface element technology. The second example problem involves creating a global-local version of the first model by graphically replacing some of the coarsely meshed components with their finely meshed counterparts. Using the IDM analysis utilities, both of these test cases were analyzed to evaluate the impact of global-local modeling. As expected, the global-local model provides a more detailed prediction of the stress distribution around the cutouts in the wing box sub-assembly. These examples serve to underscore the importance of the global-local analysis support provided by the IDM. Without accurate predictions of the stresses in critical regions, design decisions can be significantly misguided.

4.2 Conclusions and Recommendations

The baseline IDM developed in Phase I clearly demonstrates the technical feasibility of combining interface element and 3-D interactive graphics technology to form a single design environment that automates the assembly and analysis of multicomponent global-local models for faster, more accurate preliminary airframe design. The IDM's 3-D interactive design environment permits the user to graphically select, customize, and 'snap' together 'stock' components from a database to rapidly assemble airframe structural models. The design environment features powerful visualization and manipulation capabilities for dynamically positioning and resizing stock objects on the screen in real time. The IDM uses interface element technology to join components with incompatible meshes. This further reduces the engineering time by freeing the designer to connect structural components together without concern for mesh compatibility.

The IDM design environment greatly reduces the engineering time for global-local modeling. Using interface elements, coarsely meshed components in a structural model can be easily replaced (in a manner similar to 'cutting and pasting') with finely meshed versions of these components to obtain more detailed stress information in critical regions. This is significant, since inaccurate predictions of the stresses in critical regions can lead to a design whose load path characteristics exceed the material and manufacturing limitations of affordable composite structures. The global-local modeling support provided by the IDM makes it more practical to use higher-fidelity models earlier in the design cycle so that primary structures which are truly optimized for the application of affordable composites are achieved.

The relational database developed for the IDM also plays a key role in simplifying the process of model building. The process of building new airframe models has historically been very tedious and time consuming for large complex applications. The IDM database helps simplify this process by serving as a virtual warehouse of pre-meshed 'stock' aircraft components and sub-assemblies that can be used to rapidly build large multicomponent airframe models. Although new structural concepts will invariably require the use of some non-standard specialty components, once these components have been created, they can be added to the database to further expand the inventory. Once again, a key factor in the success of this approach is the interface element technology, which permits pre-meshed components to be connected together without concern for mesh compatibility.

This research provides the needed experience base and foundation for developing a next generation version of the IDM with the core features necessary to support fully associative composite aircraft design. Detailed recommendations for the development of a next generation IDM are included in the Phase II proposal.

5. ACKNOWLEDGMENT

The authors gratefully acknowledge the support of this work by the Flight Dynamics Directorate of the Air Force Wright Laboratory under contract number F33615-96-C-3206. Dr. Max Blair is the technical monitor.

6. REFERENCES

Aminpour, M. A., "A 4-Node Assumed-Stress Hybrid Shell Element with Rotational Degrees of Freedom," *NASA CR-4279*, 1990.

Aminpour, M. A., Ransom, J. B., and McCleary, S. L., "Coupled Analysis of Independently Modeled Finite Element Subdomains," Paper No. 92-2235, *Proceedings of the 33rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Dallas, TX, 1992.

Davila, C. G. and Aminpour, M. A., "Cross-Surface Interface Element for Coupling Built-Up Structural Subdomains," Paper No. 94-1336, *Proceedings of the 35th AIAA/ASME/ASCE/ AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Hilton Head, South Carolina, 1994.

Maday, Y., Mavripilis, D., and Patera, A., "Nonconforming Mortar Element Methods: Application to Spectral Discretizations," *NASA CR-181729*, 1988.

Nui, M. C. Y., "Airframe Structural Design," *Conmilit Press, Ltd.*, 1988.

Shaeffer, H. G., *MSC/NASTRAN Primer, Static and Normal Modes Analysis*, Shaeffer Analysis, Inc., Mount Vernon, New Hampshire, pp. 262-265, 1979.

Stanley, G., Hurlbut, B., Levit, Itzhak, Loden, W., Stehlin, B., and Swenson, L., "COMET-AR: COmputational MEchanics Testbed with Adaptive Refinement - User's Manual," *NASA Computational Structural Mechanics Contract Report*, Contract NAS1-18444, 1993.